

MPLAB® IDE

用户指南

请注意以下有关 Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下， Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是 "牢不可破"的。

代码保护功能处于持续发展中。 Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案 （Digital Millennium Copyright Act）》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 及位于加利福尼亚州 Mountain View 的全球总部、设计中心和晶圆生产厂均于 2003 年 10 月通过了 ISO/TS-16949:2002 质量体系认证。公司在 PICmicro® 8 位单片机、 KEELOQ® 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品方面的质量体系流程均符合 ISO/TS-16949:2002。此外， Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。



目录

第 2 部分—— MPLAB IDE 使用参考

注：



MPLAB® IDE
用户指南

前言

客户须知

所有文档均会更新，本手册也不例外。 **Microchip** 的工具和文档都在不断演变以满足客户的需求，因此在实际使用中某些对话框和 **/** 或工具的说明可能与本文档所述之内容有所不同。请访问我们的网站 （**www.microchip.com**）获取最新的可用文档。

文档均标记有 “**DS**” 编号。该编号出现在每页底部的页码之前。 **DS** 编号的命名约定为 “**DSXXXXXA**”，其中 “**XXXXX**” 为文档编号，“**A**” 为文档版本。

欲了解开发工具的最新信息，请参阅 **MPLAB®IDE** 在线帮助。在 **Help** （帮助）菜单选择 **Topics** （主题），打开现有的在线帮助文件列表。

简介

本章包含在使用 MPLAB IDE 前需要了解的一般信息。本章中讨论的内容包括：

- 文档编排
- 本指南中使用的约定
- 推荐读物
- Microchip 网站
- 开发系统客户变更通知服务
- 客户支持

文档编排

本文档说明了使用 MPLAB IDE 开发固件的方法。本手册的内容编排如下：

第 1 部分——MPLAB IDE

- **第 1 章：MPLAB IDE** 简介——讲述了 MPLAB IDE 及其如何帮助开发应用程序。
- **第 2 章：MPLAB IDE** 入门：基本教程——讲述了 MPLAB IDE v6.xx 软件的安装方法以及如何使用该软件开发示例应用程序。
- **第 3 章：步骤和教程**——逐一介绍了使用 MPLAB IDE 开发应用程序的必要步骤。每一步均给出了示例。
- **第 4 章：项目和工作区**——讲述了 MPLAB 的项目和工作区在应用程序开发过程中的使用。包括关于项目向导 （Project Wizard）、版本控制系统和项目及单文件和多文件项目的信息。

- **第 5 章：集成工具**——讲述了在 MPLAB IDE 中可使用的语言工具 （汇编器和编译器）、软件工具和硬件工具。
- **第 6 章：疑难解答**——讲述了在 MPLAB IDE 操作中的常见问题及解决方法。

第 2 部分——**MPLAB IDE 使用参考**

- **第 7 章：MPLAB IDE 界面**——讲述了 MPLAB IDE 的界面，包括菜单栏、工具栏和状态栏。
- **第 8 章：MPLAB IDE 窗口**——讲述了所有的 MPLAB IDE 窗口。包括窗口中符号的定义。
- **第 9 章：MPLAB IDE 对话框**——讲述了所有的 MPLAB IDE 对话框。
- **第 10 章：MPLAB IDE 操作参考**——讲述了关于命令行选项、快捷 （热）键、MPLAB IDE 使用的文件的综合信息以及可移植性信息。

第 3 部分——**MPLAB 编辑器**

- **第 11 章：使用编辑器**——讲述了 MPLAB 编辑器的使用方法。包括文本处理、配置编辑器、对文件和文本进行操作。本章还提供了其他补充信息，包括键盘功能、编辑器上下文相关 （鼠标右键）菜单和疑难解答。

第 4 部分——**MPLAB SIM**

- **第 12 章：软件模拟器概述**——MPLAB SIM 软件模拟器的概述。本章讨论了软件模拟器的功能、模型和执行。
- **第 13 章：MPLAB SIM 入门**——讲述了 MPLAB SIM 的初级使用。本章建议了可参考的教程并讨论了软件模拟器的功能。
- **第 14 章：使用激励**——讲述了如何对大多数 PICmicro 单片机 （MCU）和 dsPIC 数字信号控制器 （Digital Signal Controller， DSC）使用软件模拟器激励。讨论了使用 SCL 发生器创建激励的方式和激励控制。
- **第 15 章：使用激励——PIC17 器件**——详细说明了如何对 PIC17CXXX MCU 器件使用软件模拟器激励。讨论了引脚激励和文件激励。
- **第 16 章：软件模拟器疑难解答**——讲述了 MPLAB SIM 操作中的常见问题及解决方法。
- **第 17 章：软件模拟器使用参考**——详细说明了在使用软件模拟器调试应用程序的过程中可使用的功能。

本指南中使用的约定

本文档可能采用了以下约定：

文档约定

说明	表示	示例
Arial 字体：		
斜体字符	参考书目	*MPLAB IDE 用户指南*
	需强调的文字	... *仅有*的编译器 ...
首字母大写	窗口	Output （输出）窗口
	对话框	Settings （设置）对话框
	菜单选项	选择 Enable Programmer （使能编程器）
引用	窗口或对话框中的域名	“Save project before build”
带右尖括号有下划线的斜体文字	菜单路径	*File>Save*
粗体字	对话框按钮	单击 **OK**
	选项卡	单击 **Power** 选项卡
‘b*nnnn*	二进制数，其中 *n* 是一个数字	‘b00100, ‘b10
尖括号 < > 括起的文字	键盘上的按键	按下 <Enter> 或 <F1>
Courier 字体：		
常规 Courier	源代码示例	`#define START`
	文件名	`autoexec.bat`
	文件路径	`c:\mcc18\h`
	关键字	`_asm, _endasm, static`
	命令行选项	`-Opa+, -Opa-`
	位值	`0, 1`
斜体 Courier	可变参数	*`file`*`.o`，其中 *`file`* 可以是任一有效文件名
0x*nnnn*	十六进制数，其中 *n* 是一个十六进制数字	`0xFFFF`, `0x007A`
方括号 [ ]	可选参数	`mcc18` [选项] *`file`* [选项]
花括号和竖线：{ | }	选择互斥参数；“或”选择	`errorlevel {0|1}`
省略号 ...	替换重复文本	`var_name [, var_name...]`

	表示由用户提供的代码	`void main (void)`
		`{ ...`
		`}`

推荐读物

本文档介绍了使用 MPLAB IDE 的方法。下面列出了其他有用的文档。这些 Microchip 文档均已提供，建议作为补充参考资料使用。

MPLAB IDE 的自述文件 （Readme）

如需了解使用 MPLAB IDE 的最新信息，请阅读 “Readme for MPLAB IDE.txt” 文件 （ASCII 文本文件），该文件位于 MPLAB IDE 安装目录的 Readmes 子目录下。自述文件包含了本文档中可能未提供的更新信息和已知问题。

自述文件

有关使用其他工具的最新信息，请参阅与工具相关的自述文件，文件位于 MPLAB IDE 安装目录的 Readmes 子目录下。自述文件包含了本文档中可能未提供的更新信息和已知问题。

在线帮助文件

在线帮助提供了全面的 MPLAB IDE、 MPLAB 编辑器和 MPLAB SIM 软件模拟器的帮助文档。包括教程、功能说明和参考资料。

PICmicro 数据手册和系列参考手册

请访问 Microchip 网站以获取器件数据手册和相关器件系列参考手册的完整的已更新版本。

dsPIC30F Family Overview （DS70043）

dsPIC30F 器件及其架构的概述。

dsPIC30F Programmer's Reference Manual （DS70030）

该文档提供了有关 dsPIC30F 器件编程的一般信息，此外还提供了完整的指令集列表。

MICROCHIP 网站

Microchip 网站 （www.microchip.com）为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的因特网浏览器即可访问，网站提供以下信息：

- **产品支持**——数据手册和勘误表、应用笔记和范例程序、设计资源、用户指南和硬件支持文档、最新软件版本以及归档软件。
- **常规技术支持**——常见问题 （Frequently Asked Questions， FAQ）、技术支持请求、在线讨论组和 Microchip 顾问计划成员名单。
- **Microchip 的业务范围**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、 Microchip 销售办事处、代理商以及工厂代表处列表。

开发系统客户变更通知服务

Microchip 的客户通知服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲要注册，请登录 Microchip 网站 www.microchip.com，点击 “客户变更通知” 服务并按照注册说明完成注册。

开发系统产品分类如下：

- **编译器**——Microchip C 编译器及其他语言工具的最新信息。包括 MPLAB C17、MPLAB C18 和 MPLAB C30 C 编译器； MPASM™ 和 MPLAB ASM30 汇编器；MPLINK™ 和 MPLAB LINK30 目标链接器，以及 MPLIB™ 和 MPLAB LIB30 目标库管理器。
- **仿真器**——Microchip 在线仿真器的最新信息。包括 MPLAB ICE 2000 和 MPLAB ICE 4000。
- **在线调试器**——Microchip 在线调试器 MPLAB ICD 2 的最新信息。
- **MPLAB IDE**——Microchip 开发系统工具的 Windows® 集成开发环境 MPLAB IDE 的最新信息。主要针对 MPLAB IDE、MPLAB SIM 软件模拟器、MPLAB IDE 项目管理器以及一般编辑和调试功能。
- **编程器**——Microchip 编程器的最新信息。包括 MPLAB PM3 和 PRO MATE® II 器件编程器以及 PICSTAR® Plus 开发编程器。

客户支持

Microchip 产品的用户可以通过以下渠道获取帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师 （FAE）
- 技术支持
- 开发系统信息热线

用户应联系代理商、代表处或应用工程师 （FAE）获取支持。当地销售办事处也可为用户提供帮助。本文档后附有销售办事处及其地址的列表。

也可通过 http://support.microchip.com 网站获取技术支持。

注：



第 1 部分——MPLAB IDE

注：



第 1 章 MPLAB® IDE 简介

1.1 嵌入式系统简介

MPLAB IDE 是一种在 PC 机上运行的软件，用来为 Microchip 单片机开发应用程序。由于它提供了一种单一的集成环境来为嵌入式单片机开发程序代码，因此被称为集成开发环境或 IDE。经验丰富的嵌入式系统设计人员可以跳过这部分而直接转至**第 1.7 节 "MPLAB IDE 的组件"**。在此还建议浏览**第 1.9 节 "MPLAB IDE 在线帮助"**和**第 1.10 节 "MPLAB IDE 更新和版本编号"**。本章的其余部分简要地介绍了嵌入式系统开发以及使用 MPLAB IDE 的方法。

1.1.1 "嵌入式系统"说明

嵌入式系统通常为一种利用小型单片机 （如 Microchip PICmicro® MCU 或 dsPIC® 数字信号控制器 （DSC））功能的设计。这种单片机将微处理器 （类似于台式计算机中的 CPU）与某些称为 "外设"的附加电路相结合，加上同一芯片上的其他电路就构成了一个需要极少其他外部器件的小型控制模块。这种单一的器件可以被嵌入到其他电子和机械设备中，以实现低成本的数字控制。

1.1.2 嵌入式控制器和 PC 机之间的区别

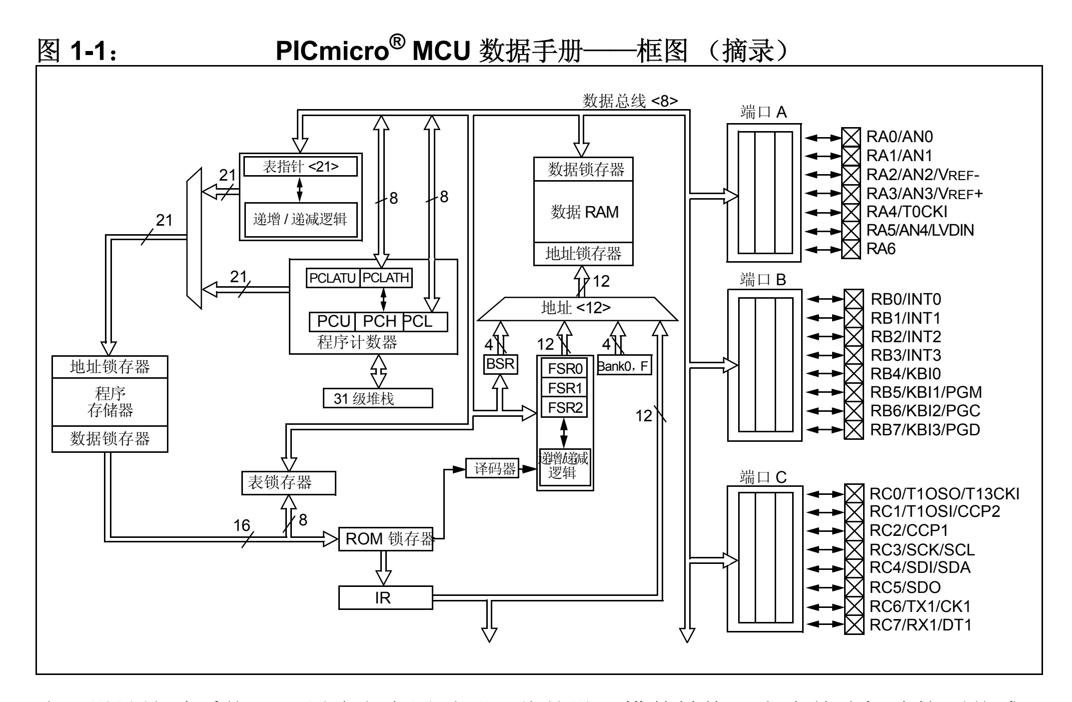
嵌入式控制器与 PC 机之间的主要区别在于：嵌入式控制器专用于某项特定任务，或某类特定任务。而 PC 机则设计为能够运行多种不同类型的程序，并能够连接到多种不同的外部设备。嵌入式控制器只有一个程序，因此成本低廉，只要能够保证处理专项任务所需的计算能力和硬件即可。而 PC 机的核心具有比较昂贵的通用中央处理单元 （CPU），它还包含了很多其他外部设备 （内存、磁盘驱动器、视频控制器和网络接口电路等）。嵌入式系统具有低成本的智能型微控制器单元 （MCU），在同一芯片上还有很多外设电路而外部设备则相对较少。通常，嵌入式系统属于隐藏部件，或者是其他产品的子模块，如电池式钻孔机、冰箱或车库开门器。此类产品中的控制器只执行整个设备的一小部分功能。控制器给这些设备中的关键子系统增添了低成本的智能。

举例来说，烟雾检测器就是一种嵌入式系统。它的功能是检测传感器输出的信号，如果信号表示有烟雾存在，则发出警报。可以使烟雾检测器中的小程序执行无限循环，不停地对烟雾传感器输出的信号进行采样，也可以使烟雾检测器处于低功耗的 "休眠"模式，由传感器的输出信号将它唤醒。唤醒后烟雾检测器中的小程序就会发出警报。该程序可能还具有一些其他功能，如用户测试功能和电池欠压报警。虽然配备传感器和音频输出设备的 PC 机通过编程也可以实现上述功能，但这并不是一种节约成本的解决方案 （而且它也不可能在 9 伏的电压下以无人照管方式常年运作！）。嵌入式设计使用廉价的单片机，为我们的日常生活环境提供智能化处理，如烟雾检测器、相机、手机、家用电器、汽车、智能卡以及安防系统。

1.1.3 单片机的组件

PICmicro MCU 具有用于储存固件或编码指令的程序存储器，以运行程序。它还具有“文件寄存器”存储器，用来储存程序计算或临时数据所需的变量。单片机芯片本身还具有很多外围设备。其中某些外设被称为 I/O 端口。 I/O 端口是单片机上的引脚，可以将其驱动为高电平或低电平以发送信号、闪烁指示灯或驱动扬声器——发送任何能够通过线路传输的信息。这些引脚通常为双向引脚，可以配置为输入，以使程序能够对外部开关或传感器作出响应，或是与某些外部器件通信。

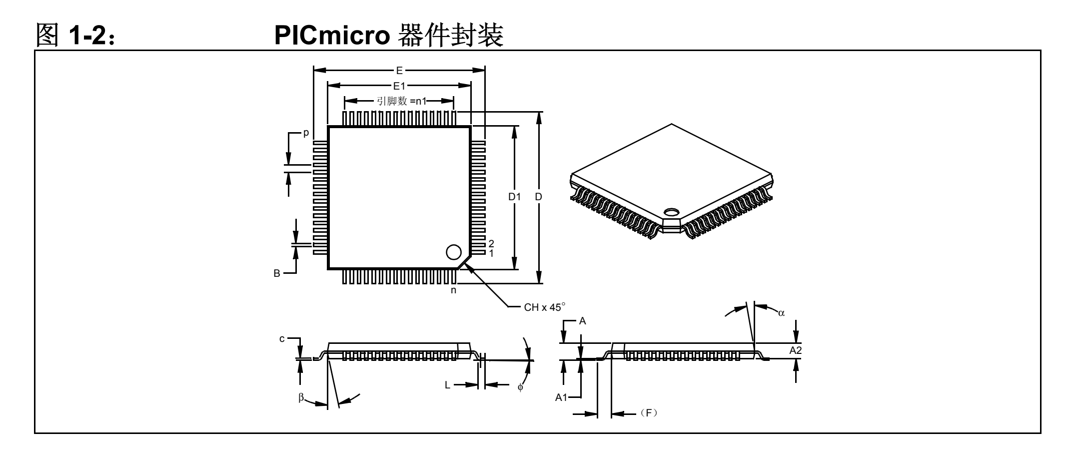
图 1-1： PICmicro® MCU 数据手册——框图 （摘录）



为了设计这种系统，必须确定应用需要哪些外设。模数转换器允许单片机连接到传感器并接收变化的电平。串行通信外设可以通过线缆以串行方式与另一台单片机、局域网或因特网通信。 PICmicro MCU 上称为定时器的外设可以精确的检测信号事件并生成和捕捉通信信号以及输出精确的波形，甚至可以在单片机挂起或因电源故障或硬件故障导致单片机工作异常时自动将其复位。其他外设可以检测外部电源是否降到了危险电平之下，以便让单片机能够及时存储重要信息，从而在完全掉电之前安全关闭。

应该使用哪种 PICmicro MCU 在很大程度上取决于运行应用程序所需的外设和存储器大小。其他因素可能包括单片机的功耗及其 “外形因素”，即安装到目标设计中的物理封装尺寸和特性。

图 1-2： PICmicro 器件封装



1.1.4 使用 MPLAB IDE 实现嵌入式系统设计

嵌入式控制器开发系统是在台式计算机上运行的程序系统，它有助于编写、编辑和调试程序代码 （嵌入式系统应用的灵魂 ），并将其烧写到单片机中。 MPLAB IDE 在 PC 机上运行，包含设计和部署嵌入式系统应用所需的全部组件。

开发嵌入式控制器应用的典型任务有：

1. 创建高端设计。根据所需的功能和性能，决定最适用于应用的 PICmicro 或 dsPIC 器件，然后设计相关的硬件电路。在决定由哪些外设和引脚控制硬件之后，编写固件 （控制嵌入式应用中的硬件的软件）。可以使用语言工具编写和编辑代码，这些语言工具如汇编器 （可以直接将汇编代码转换为机器码）或编译器 （允许使用更通用的语言创建程序）。汇编器和编译器允许使用功能标号来标识代码子程序，变量名可与其用途相关联，并采用有助于在可维护的结构中组织代码的程序结构，从而使代码易于理解。

图 1-3： PICmicro MCU 数据手册——时序 （摘录）

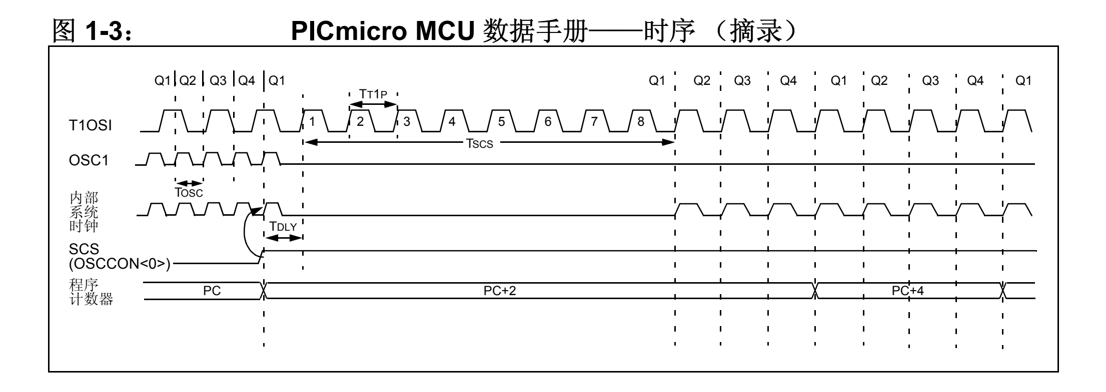
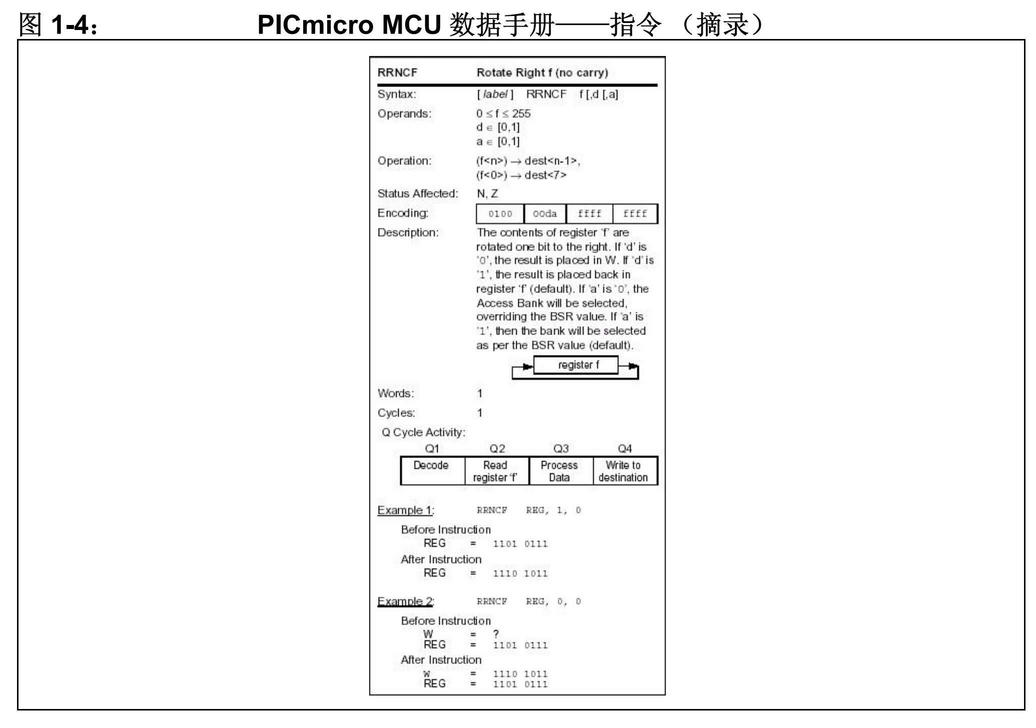


图 1-4： PICmicro MCU 数据手册——指令 （摘录）



RRNCF **Rotate Right f (no carry)**

Syntax: [*label*] RRNCF f [,d [,a]

Operands: $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

Operation: (f<n>) → dest<n-1>,
(f<0>) → dest<7>

Status Affected: N, Z

Encoding:

0100	00da	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the right. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' is '1', then the bank will be selected as per the BSR value (default).

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example 1: RRNCF REG, 1, 0

Before Instruction
REG = 1101 0111
After Instruction
REG = 1110 1011

Example 2: RRNCF REG, 0, 0

Before Instruction
W = ?
REG = 1101 0111
After Instruction
W = 1110 1011
REG = 1101 0111

2. 使用汇编器和 / 或编译器以及链接器汇编、编译和链接软件以将您的代码转换为 “0 和 1 序列”——可被 PICmicro MCU 识别的机器码。机器码最终将变为固件 （编程到单片机中的代码）。
3. 测试代码。通常，复杂的程序不一定会按照预期运行，要得到正确的结果，还需要除去设计中的 “错误”（bug）。您可以通过调试器观察与所编写的带有符号和函数名的源代码相对应的机器码中 “0 和 1 序列” 的执行。在调试过程中，您可以测试代码以观察变量在程序执行过程中各个点的值、进行 “what if” 检查、更改变量值和单步调试程序。
4. 将代码 “烧写” 到单片机中，验证其在最终的应用中是否能正确执行。

当然，其中的每个步骤都可能非常复杂。重要的是必须关注设计中的细节，并依靠 MPLAB IDE 及其组件来完成每个步骤，这样就不会不断地将时间浪费在学习上。

虽然可使用 MPLAB IDE 对电路和代码进行建模，以便做出关键的设计决定，步骤 1 仍需由设计人员完成。

MPLAB IDE 真正起帮助作用的是在步骤 2 到步骤 5。它的程序编辑器有助于使用选定的语言工具编写正确的代码。编辑器可以识别汇编器和编译器的编程语法结构，从而自动将源代码以不同颜色区分，这有助于确保代码在语法上的正确性。项目管理器有助于组织应用程序中使用的各种文件：源文件、处理器描述头文件以及库文件。编译了代码之后，您还可以控制编译器以何种程度优化代码大小或执行速度，以及将在器件中的哪些部分存储各个变量及程序数据。您也可以指定 “存储器模型”以使您的应用能最佳地利用单片机的存储器。如果在编译应用程序时语言工具报错，则会显示出错的行，双击它即可转到对应的源文件，以便立即编辑。编辑后，可以按 “build” （编译）按钮再次尝试。由于要编写和测试很多子程序段，因此复杂的代码通常会经过许多次这样的编写—编译—修正过程。 MPLAB IDE 会以最快的速度执行这一过程，从而使您能够尽快转入下一个步骤。

代码编译没有错误之后，还需要对其进行测试。MPLAB IDE 具有称为 “调试器”的组件和免费的软件模拟器，以帮助所有的 PICmicro 和 dsPIC 器件测试代码。即使当硬件还没有完成时，您也可以使用软件模拟器开始测试代码。软件模拟器就是一种模拟单片机执行的软件。软件模拟器可以接收模拟输入 （激励信号），以便模拟固件对外部信号的响应。软件模拟器可以测试代码执行时间、单步调试代码以观察变量和外设，并跟踪代码以生成详细的程序运行记录。

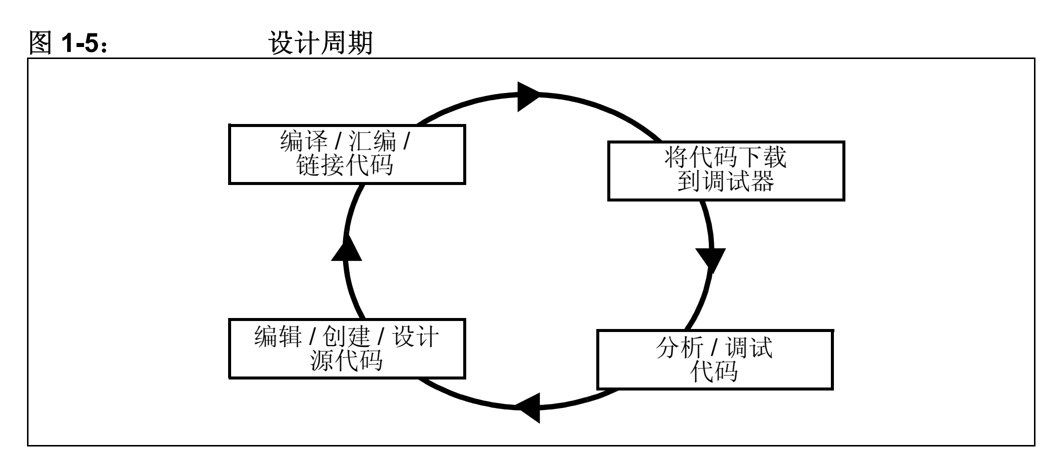
一旦硬件进入样机阶段，就可以使用诸如 MPLAB ICE 或 MPLAB ICD 2 的硬件调试器了。这些调试器在实际的应用上实时运行代码。 MPLAB ICE 实际上取代了目标板上的单片机，它使用高速仿真头对设计中的硬件进行完全控制。 MPLAB ICD 2 则使用在许多带有闪存程序存储器的 Microchip MCU 中内置的特殊电路，并且能 “检查”目标单片机中的程序和数据存储器。 MPLAB ICD 2 可以停止和开始执行程序，使您可以直接使用应用板上的单片机测试代码。

应用程序正确运行之后，就可以使用某一 Microchip 器件编程器来对单片机编程了，如 PICSTART Plus 或 MPLAB PM3。这些编程器可以验证将要执行的最终代码和所设计的是否一致。 MPLAB IDE 支持大多数 PICmicro MCU 和所有的 dsPIC 数字信号控制器。

1.2 开发周期

编写应用程序的过程通常被称为开发周期——因为第一次就可以完成从设计到实现的所有步骤而不出任何差错是很少的。通常，编写代码之后要进行测试和修改才能生成正确执行的应用程序。集成开发环境让嵌入式系统设计工程师能够顺利完成这种开发周期，而不会因为要在各种工具之间切换而分心。使用 MPLAB IDE，所有功能都集成在一起，工程师就可以专心完成应用程序，而不会因为要切换不同的工具和操作模式而中断开发。

图 1-5： 设计周期

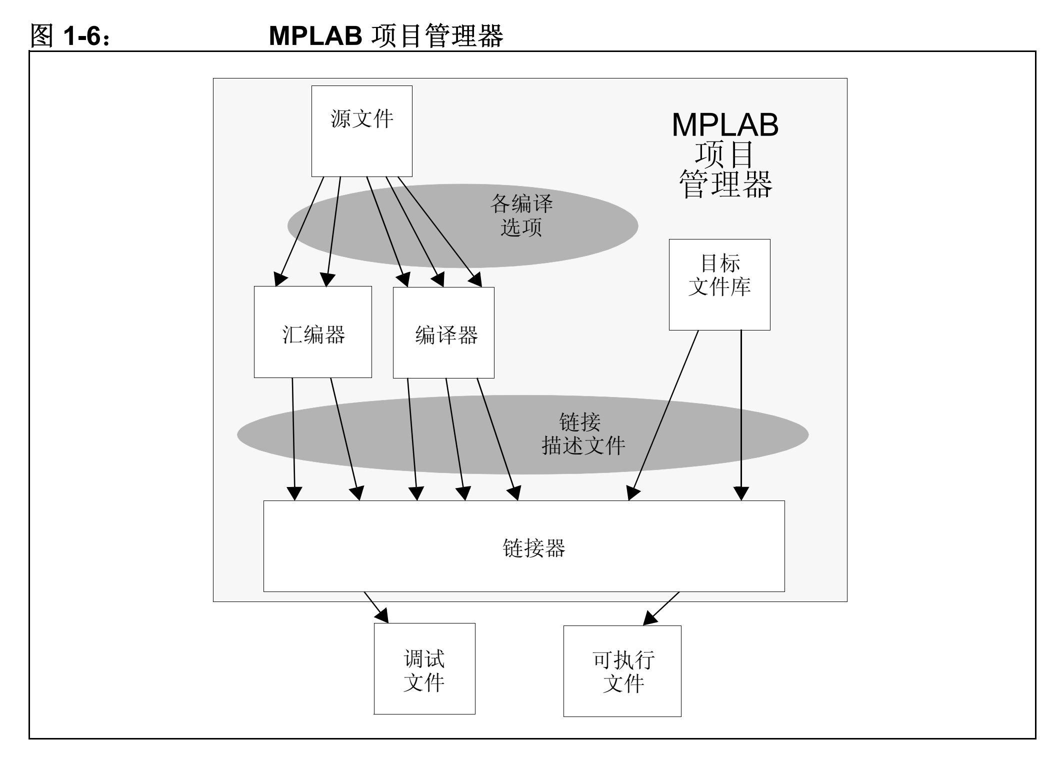


IDE 就是一种 “包装器”（wrapper），它协调同一图形用户界面的所有工具——这一过程通常是自动完成的。例如，一旦代码编写完成，就可以将其转换为可执行指令，并下载到单片机中观察它的运行。这一过程需要多种工具：编写代码的编辑器、管理文件及设置的项目管理器、将源代码转换为机器码的编译器或汇编器，以及多种连接到目标单片机的硬件或用来模拟单片机运行的软件。

1.3 项目管理器

项目管理器管理要编辑的文件和其他相关文件，以便将这些文件送至语言工具进行汇编或编译，并最终送至链接器。链接器的任务是将来自汇编器、编译器和库的目标代码片段存放到嵌入式控制器的恰当存储区，并确保各个模块之间可以相互协调（或"链接在一起"）。从汇编、编译到链接的整个操作过程被称为项目"build"。使用 MPLAB 项目管理器，每个文件可以根据需要调用不同的语言工具。并且编译过程会将所有的语言工具操作集成到一起。

图 1-6： MPLAB 项目管理器



源文件是遵从汇编器或编译器规则编写的文本文件。汇编器和编译器将源文件转换为中间模块机器码和占位符，以作为函数和数据存储的参考。链接器解析这些占位符，并将所有模块合并为一个可执行的机器码文件。链接器还会生成一个调试文件，允许 MPLAB IDE 将正在执行的机器码与源文件相关联。

文本编辑器用于编写代码。这不是普通的文本编辑器，而是专门为编写 Microchip MCU 的代码而设计的编辑器。它可以识别文本中的语法结构，并采用彩色编码来区分各种元素，如指令助记符、 C 语言结构和注释。编辑器支持编写源代码常用的操作，如查找 C 语言中的匹配花括号、注释和取消注释代码段、查找多个文件中的文本以及添加特殊书签等。编写好代码之后，编辑器可以配合其他工具，显示调试器中代码的执行。编辑器中可以设置断点，而且将鼠标指针悬停在变量名上方还可以查看变量的值。可以将源代码窗口中的变量名拖拽到 Watch （观察）窗口中。

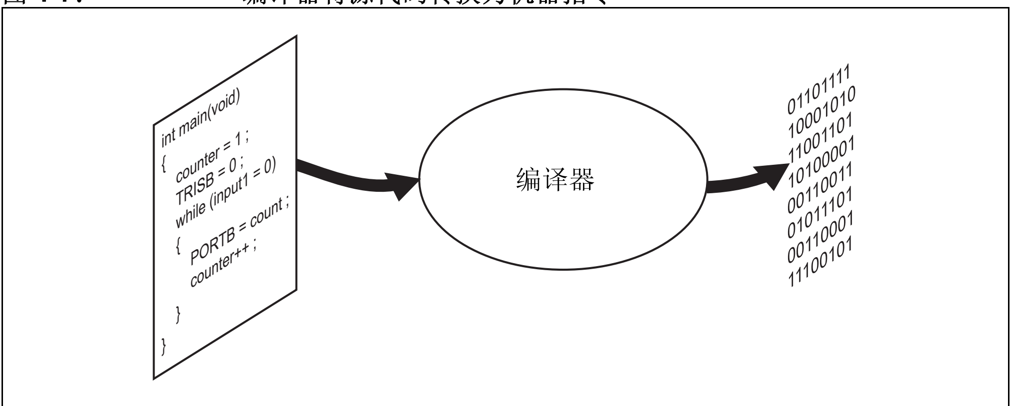
1.4 语言工具

语言工具就是诸如交叉汇编器和交叉编译器的程序。大多数人都比较熟悉在 PC 机上运行的语言工具，比如 Visual Basic 或 C 编译器。当使用嵌入式系统的语言工具时，就会使用 “交叉汇编器” 或 “交叉编译器”。这些工具与常见编译器的区别在于，它们在 PC 机上运行，但生成的代码却在其他微处理器 （单片机）上运行，即它们为单片机交叉编译代码，单片机使用与 PC 机完全不同的指令集。

语言工具还会生成调试文件， MPLAB IDE 使用这个文件将机器指令和存储单元与源代码相关联。这种集成让 MPLAB 编辑器能够设置断点，允许在 watch 窗口中查看变量的内容，并允许您单步调试源代码，观察应用程序的执行。

嵌入式系统语言工具与在 PC 机上运行和执行的编译器还有一点不同，那就是对空间非常敏感。生成的代码量越小越好，因为这样可以尽可能的减少目标对存储器的占用，从而降低成本。这就意味着需要一种使用特定于机器的知识来优化和增强代码的技术。对于 PC 机，复杂程度适中的程序通常就会大到数兆字节。而简单的嵌入式系统程序则可以小至几千字节，甚至更小。中等规模的嵌入式系统可能需要 32K 或 64K 的代码，以实现相对复杂的功能。某些嵌入式系统会使用数兆字节的空间，以存储大型表、用户文本消息或数据日志。

图 1-7： 编译器将源代码转换为机器指令



1.5 目标调试

在开发环境中，代码的执行是在调试器中测试的。调试器可以是软件程序，用来模拟单片机的操作以便进行测试，也可以是特殊的仪器，用来分析在应用程序中执行的代码。

MPLAB IDE 中内置了软件模拟器，因此不需要任何其他硬件就可以对程序进行测试。软件模拟器是一种软件调试器，软件模拟器的调试功能与硬件调试器的功能几乎完全相同，学习起来非常容易。由于软件模拟器使用 PC 机中的 CPU 来模拟单片机的操作，因此它通常比实际单片机运行得要慢一些。 MPLAB IDE 为所有的 PICmicro 和 dsPIC 处理器提供了许多软件模拟器。

在 MPLAB IDE 中，您可以使用两种类型的硬件：编程器和硬件调试器。编程器直接将机器码从 PC 机传送到目标单片机的片内存储器中。然后就可以将单片机插入到应用中了，并希望单片机中的程序能够按设计运行。

但是，代码通常都不会完全按照预期运行，设计人员需要检查代码及其在应用中的运行状况，以决定如何修改源代码，使之能够按预期运行。这个过程被称为调试。如前所述，软件模拟器可以用于测试代码的运行，但是一旦向单片机烧写了固件，就由软件模拟器之外的很多因素发挥作用了。仅使用编程器，虽然可以更改代码，并将其重新烧写到单片机，然后插入目标进行重新测试，但如果代码比较复杂，这个过程就非常费时费力，而且很难弄清楚硬件中究竟发生了什么问题。

在这种情况下，硬件调试器就非常有用了。硬件调试器可以是在线仿真器，它使用专用硬件来代替实际的目标单片机。它也可以是在线调试器，使用具有特殊内置调试功能的单片机。硬件调试器和软件模拟器一样，允许设计人员检查当代码执行到各个点时的变量，单步调试程序以检查硬件与专用电路交互时的指令。

在项目设计周期的末期，调试通常会变得紧迫。随着交付期的临近，让应用程序按最初设计运行是部署产品之前的最后一步，这通常是使产品交付延期的最大因素。这就是集成开发环境最重要之处。调试和修改代码、重新编译、下载和测试——所有都需要时间。在一个环境中使用所有工具可以减少该 “周期”的时间。能否在最后这几个步骤中找出关键错误对嵌入式系统设计人员是一种考验。使用正确的工具可以节省时间。使用 MPLAB IDE 可以选择多种工具，不过这些工具的界面都是类似的，因此从软件模拟器到低成本的在线调试器再到强大的在线仿真器的学习过程也会比较简单。

1.6 对器件编程

应用程序经过调试并在开发环境中运行后，还需要对其进行测试。可以使用在线调试器或器件编程器对器件进行编程。 MPLAB IDE 可以设置为编程器功能，从而为器件烧写程序。目标应用程序现在可视为接近完成的状态。样机开发编程器可以快速制作和评估样机。某些应用程序可以在器件焊接到目标板上之后进行编程。使用在线串行编程 （In-Circuit Serial Programming™ ， ICSP™），可以在生产过程中将固件编程到应用中，从而能够在嵌入式应用开发的末期将更新的版本编程到嵌入式应用中。支持在线调试的器件甚至可以在生产完成后重新插入到 MPLAB ICD 2 中，以进行质量检测和下一代固件的开发。

1.7 MPLAB IDE 的组件

MPLAB IDE 具有内置组件和插入模块来为系统配置各种软件和硬件工具。

1.7.1 MPLAB IDE 内置组件

内置组件包括：

- **项目管理器**

项目管理器提供 IDE 和语言工具之间的集成和通信。

- **编辑器**

编辑器是功能全面的程序文本编辑器，它还可以作为调试器的窗口使用。

- **汇编器 / 链接器和语言工具**

汇编器可以单独使用来汇编单个文件，也可以与链接器配合使用，以从不同的源文件、库和重新编译的目标文件生成一个项目。链接器负责将编译后的代码分配到目标单片机的存储区中。

- **调试器**

Microchip 调试器允许使用断点、单步调试、 watch 窗口以及目前 MPLAB IDE 调试器中的所有功能。将它与编辑器配合使用，可以将正在调试的目标中的信息与源代码相关联。

- **执行引擎**

MPLAB IDE 中具有适用于所有 PICmicro 和 dsPIC 器件的软件模拟器。这些软件模拟器使用 PC 机来模拟 PICmicro 和 dsPIC 器件的指令和某些外设功能。还可以使用可选的在线仿真器和在线调试器通过在应用硬件中运行代码来测试代码。

1.7.2 MPLAB IDE 的其他可选组件

您可以选购以下组件并将它们添加到 MPLAB IDE 中：

- 编译器语言工具

Microchip 的 MPLAB C17、 MPLAB C18 和 MPLAB C30 可以提供高度集成的优化代码。MPLAB IDE 项目管理器可以调用这些编译器以及 HI-TECH、 IAR、 microEngineering Labs、 CCS 和 Byte Craft 生产的编译器，来编译自动载入到目标调试器中的代码，进行即时测试和校验。

- 编程器

PICSTART Plus、PRO MATE II、MPLAB PM3 以及 MPLAB ICD 2 都可以将代码编程到目标单片机中。 MPLAB IDE 提供了对代码和数据编程的完全控制，还可以使用 MPLAB IDE 对配置位进行设置来设定目标单片机的各种操作模式。

- 在线仿真器

MPLAB ICE 2000 和 MPLAB ICE 4000 是功能全面的适用于 PICmicro 和 dsPIC 器件的仿真器。它们通过 I/O 端口与 PC 机相连，可以对目标应用中的单片机操作进行完全控制。

- 在线调试器

MPLAB ICD 2 提供了一种取代仿真器的经济方案。通过使用部分片上资源， MPLAB ICD 2 可以将代码下载到被插入到应用中的目标单片机中，并设置断点，单步调试和监视寄存器及变量。

1.8 MPLAB IDE 文档

现有以下文档可帮助您使用 MPLAB IDE：

- *MPLAB IDE Quick Chart* （DS51410）
- *《MPLAB IDE 快速入门指南》* （DS51281C_CN）
- *《MPLAB IDE 用户指南》*（DS51519A _CN）

与 MPLAB IDE 配合使用的各种 Microchip 软件和硬件工具也存在相应文档。请访问 Microchip 网站获取这些文档的可下载 pdf 版本。

1.9 MPLAB IDE 在线帮助

由于 MPLAB IDE 还在不断地更改中 （请参阅第 **1.10 节 “MPLAB IDE 更新和版本编号”**），本文中提到的部分细节可能会有所改变。对话框的外观可能与本手册中介绍的不完全一致，菜单列表的顺序也可能不同，或是增加了新的项。因此，在线帮助是您当前使用的 MPLAB IDE 版本的最佳参考。

MPLAB IDE 具有大量的在线帮助，处在不断更新之中。如果在使用 MPLAB IDE 时遇到了问题，一定要查看在线帮助寻找答案。最重要的是，在线帮助列出了特定工具在支持特定器件时可能存在的任何限制。在使用新的器件 / 工具组合之前，请始终尝试查看这部分内容。

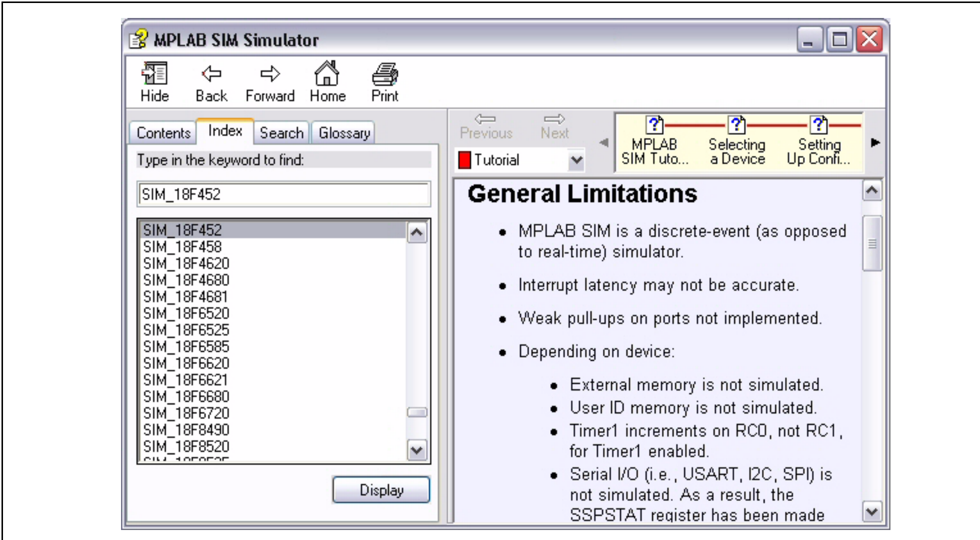
Debugger>Settings 对话框的 “Limitations”（限制）选项卡给出了软件模拟器、仿真器或在线调试器与正在被模拟的实际器件相比可能存在的任何限制。一般限制则显示在文本区中。

图 1-8： *DEBUGGER>SETTINGS*：LIMITATIONS 选项卡

按下 **Details** （详情）按钮，显示与要被调试的器件相关的具体限制。从这里显示的信息中还可以看到与调试器相关的一般限制。

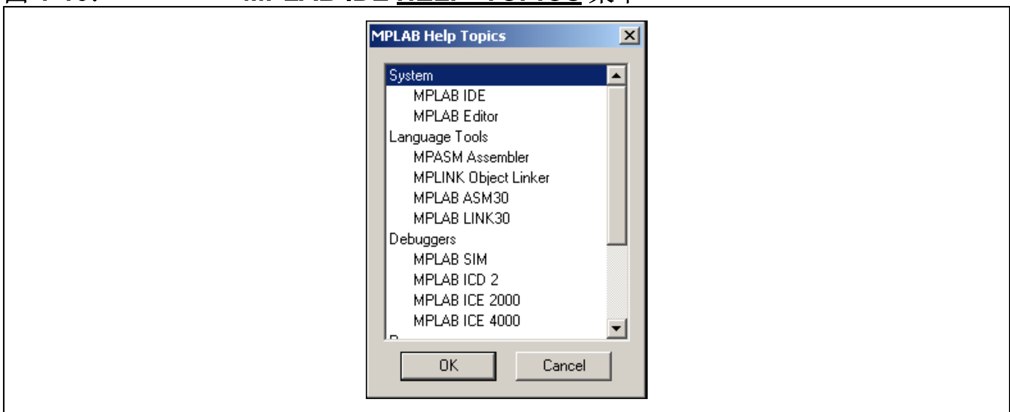


图 1-9： 软件模拟器限制详细信息



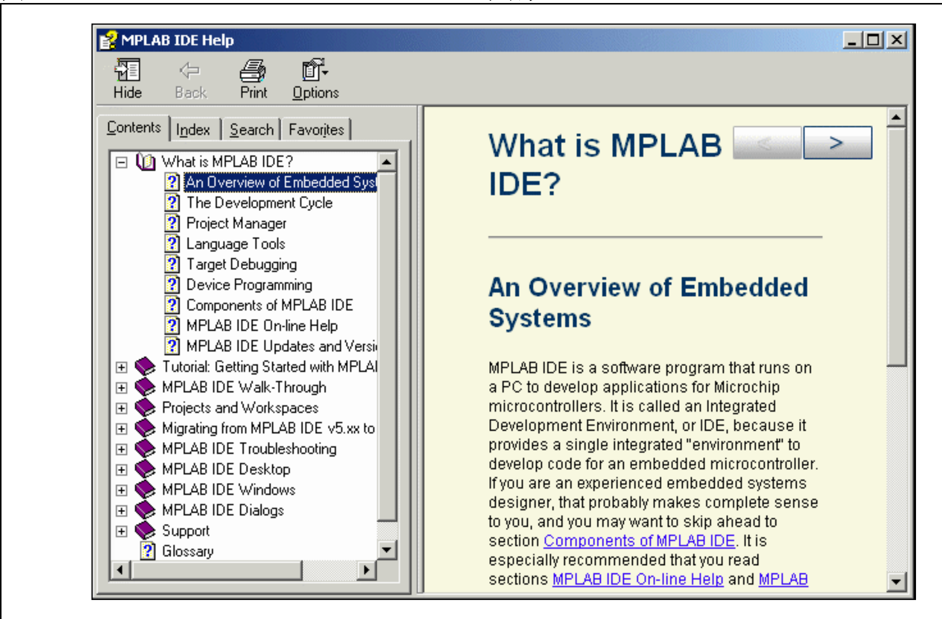
从 MPLAB IDE 的 Help （帮助）菜单，选择 *Help>Topics* ，获取 MPLAB IDE 及其所有组件的帮助列表。

图 1-10： MPLAB IDE *HELP>TOPICS* 菜单



MPLAB IDE 帮助涵盖了 MPLAB IDE 的各个方面以及所有的 Microchip 工具。可以按目录或索引方式离线查看它，也可以使用搜索实用程序查找有关任何 MPLAB IDE 主题的帮助。MPLAB IDE 帮助还可以指引用户获取其他类型的帮助，如 Microchip Update Notification system （Microchip 更新通知系统）。

图 1-11： MPLAB IDE HELP 对话框



1.10 MPLAB IDE 更新和版本编号

MPLAB IDE 是一个不断改进的程序，具有成千上万的用户。 Microchip 在不断地设计具有新功能的新型单片机。许多新的 MPLAB IDE 功能都源自客户的需求和内部使用。不断开发的新设计和发布的新单片机促使 MPLAB IDE 将不断改进。 MPLAB IDE 计划大约每四个月更新一次，以便增加新的器件支持和新的功能。此外，在两个主要的发行版之间，还会发布另外的 "临时" 发行版。 MPLAB IDE 的版本编号方案可以反映出当前版本属于主要产品发行版还是临时发行版。如果版本号以零结束，即 MPLAB IDE v6.50、 v.6.60 或 v7.00，则表示该版本是主要的产品发行版。如果版本号以零以外的数字结束，即 v6.41、 v6.52 或 v7.55，则表示该版本是临时发行版。临时发行版主要是为了提供对新器件或组件的早期试用，或进行快速重要修正及新功能的预演。这种临时发行版并没有经过如产品发行版那样的完整测试，因此建议不要在严格的设计中使用这种版本。我们建议在开发过程中使用产品发行版，除非是在使用了新器件或组件，或是遇到了临时发行版中已修正的某个问题时，才使用临时发行版以有效利用 MPLAB IDE。

MPLAB IDE 软件的每一次新的发布都实现了新的功能，因此印刷文档的内容肯定会比在线帮助滞后。在线帮助是解决有关 MPLAB IDE 中任何问题的最好参考。

要在 MPLAB IDE 及其组件更新时得到通知，请在 www.microchip.com 网站上订阅 "客户变更通知" 服务的 " Development Tools" 部分。



第 2 章 MPLAB IDE 入门：基础教程

2.1 简介

MPLAB 集成开发环境 （IDE）是一个综合的编辑器、项目管理器和设计平台，适用于使用 Microchip PICmicro 和 dsPIC 单片机进行嵌入式设计的应用开发。

本章讲述了 MPLAB IDE 的初步使用。本简短教程将讲述如何创建项目、编辑代码和测试应用程序。学习完本教程后，就可以快速了解项目管理器、编辑器和调试器的基本概念。在后面的章节中还会讲述 MPLAB IDE 的完整功能。

本节将详细说明 MPLAB IDE 的安装和卸载。接下来是介绍创建项目和说明 MPLAB IDE 基本调试功能的简单逐步教程。学习完教程后， MPLAB IDE 的新手将对使用此系统开发应用有一个基本的了解。无需前面的知识，以及 MPLAB IDE 及其组件的综合技术细节，就可形成使用 MPLAB IDE 的基本框架。

本教程将涉及以下基本步骤：

- MPLAB IDE 的功能和安装
- 教程概述
- 选择器件
- 创建项目
- 设置语言工具
- 为项目命名
- 创建代码
- 再次编译项目
- 使用软件模拟器测试代码
- 教程总结

2.2 MPLAB IDE 的功能和安装

MPLAB IDE 是基于 Windows® 操作系统的集成开发环境，适用于 PICmicro MCU 系列和 dsPIC 数字信号控制器的开发。 MPLAB IDE 提供以下功能：

- 使用内置的编辑器创建和编辑源代码。
- 汇编、编译和链接源代码。
- 通过使用内置的软件模拟器观察程序流程，或者使用在线仿真器或在线调试器以实时方式观察程序流程来调试可执行逻辑。
- 用软件模拟器或仿真器进行时序测量。
- 查看 Watch 窗口中的变量。
- 使用器件编程器将固件烧写入器件 （欲知详细信息，请查阅具体的器件编程器的用户手册）。

注：	MPLAB IDE 也支持部分第三方工具。详情请查阅发行声明或自述文件。

2.2.1 安装 / 卸载 MPLAB IDE

要在系统上安装 MPLAB IDE：

注：	对于某些 Windows 操作系统，需要管理员权限才能在个人计算机上安装软件。

- 若想从光盘安装，请将安装光盘放入光驱。遵循屏幕上的菜单来安装 MPLAB IDE。若未出现屏幕菜单，请使用 “Windows 资源管理器”进行查找并执行 CD-ROM 中的可执行文件 `menu.exe`。
- 若想从 Microchip 网站 （www.microchip.com）下载 MPLAB IDE，请找到下载（.zip）文件，选择该文件然后将它保存到计算机。将该文件解压缩并执行解压缩后的文件来进行安装。

要卸载 MPLAB IDE：

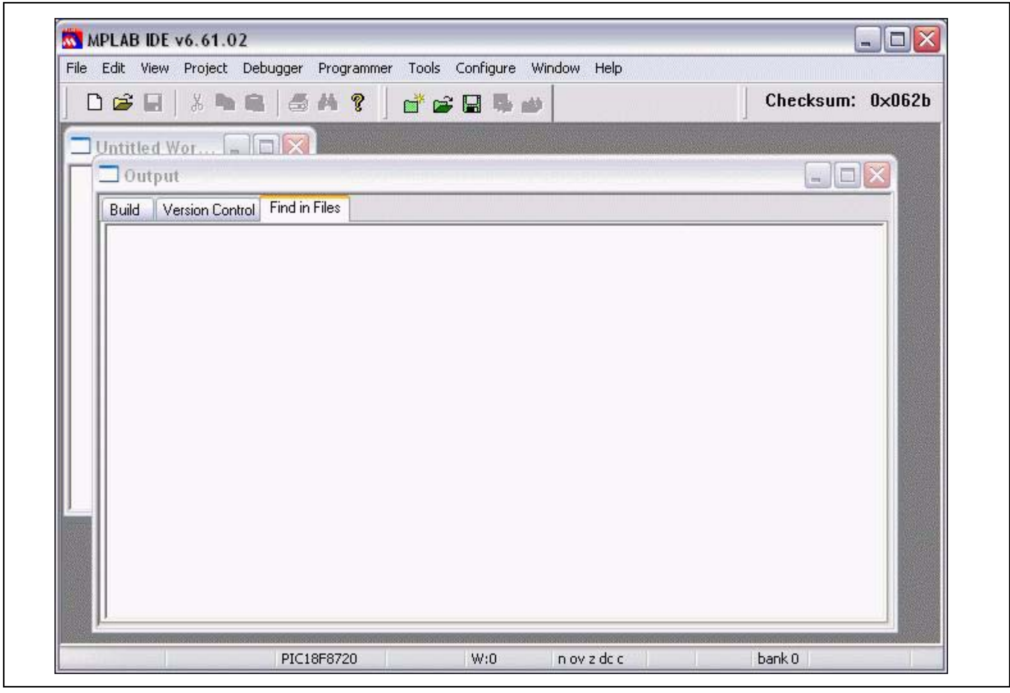
- 选择*开始 > 设置 > 控制面板*，打开控制面板。
- 双击 “添加 / 删除程序”。在列表中找到 MPLAB IDE 并单击它。
- 单击**更改 / 删除**将该程序从系统中删除。

2.2.2 运行 MPLAB IDE

软件安装后，要启动 MPLAB IDE，双击桌面上的图标或选择 *Start>Programs>Microchip MPLAB IDE vx.x>MPLAB IDE vx.x*。屏幕会显示 MPLAB IDE 徽标并随后显示 MPLAB IDE 界面 （图 2-1）。



图 2-1： MPLAB IDE 界面



2.3 教程概述

要创建目标 PICmicro MCU 可执行的代码，需要将源文件添加到项目中，然后可以使用选定的语言工具 （汇编器、编译器和链接器等）将代码编译为可执行代码。在 MPLAB IDE 中，由项目管理器控制这一过程。

所有项目均具有以下基本步骤：

- 选择器件

  MPLAB IDE 的功能根据所选择的器件而有所不同。应该在开始项目之前完成器件选择。
- 创建项目

  将使用 MPLAB 项目向导来创建项目。
- 选择语言工具

  将在项目向导中选择语言工具。就本教程而言，将使用内置的汇编器和链接器。而在其他项目中，也许需要选择 Microchip 的某一编译器或其他第三方工具。
- 将文件添加到项目

  将在项目中添加两个文件，一个模板文件和一个链接描述文件。这两个文件都位于 MPLAB IDE 文件夹的子文件夹中。使用这两个文件，开始项目就容易多了。
- 创建代码

  将向模板文件添加一些代码，从 I/O 端口发出一个递增的值。
- 编译项目

将编译项目——使源文件被汇编和链接成可以在选定的 PICmicro MCU 上运行的机器码。

• 使用软件模拟器测试代码

最后，将使用软件模拟器测试代码。

项目向导将轻松地指导我们完成大部分的步骤。

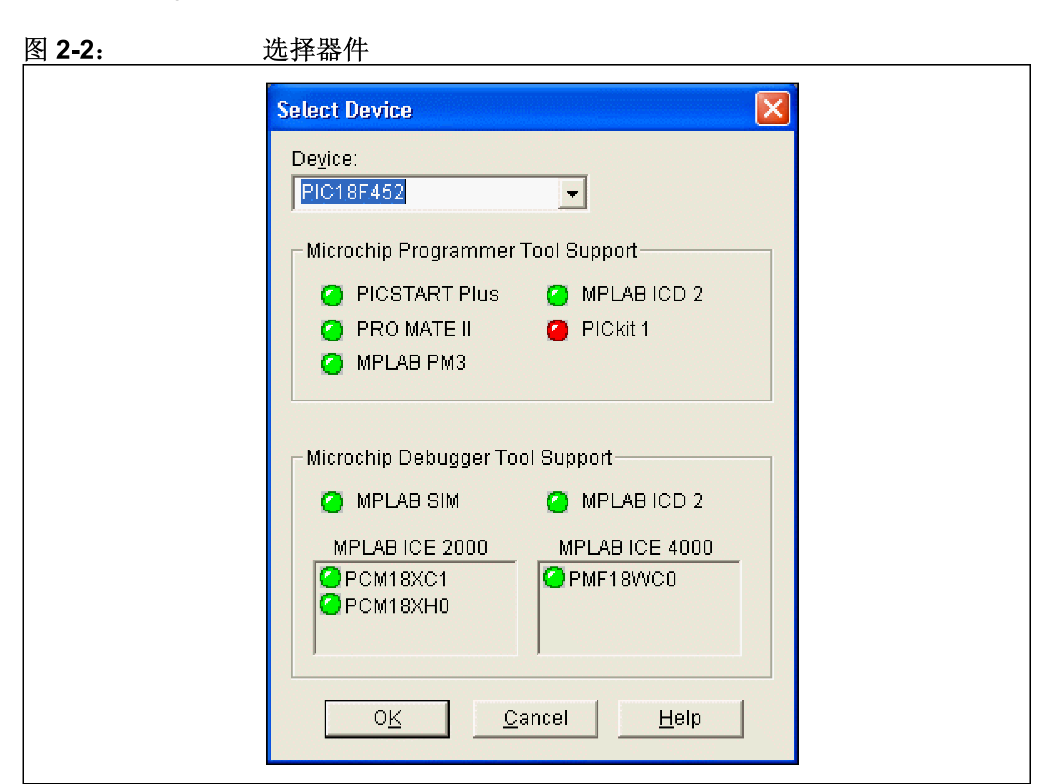
注：	在今后的产品发布中用户界面中的某些部分可能会有所更改，并且本教程中的屏幕截图可能也会与以后发布的 MPLAB IDE 界面不完全相符。随着更多器件的发布，将会添加新的功能。本教程中描述的任何功能均不会被删除，并且可能添加更多的功能。在线帮助是当前版本的 MPLAB IDE 最新的参考资料。

2.4 选择器件

为了在本文档中显示菜单选择的过程， MPLAB IDE 中顶行菜单中的菜单项，将会显示在菜单名之后，如 *MenuName>MenuItem*。要在 *Configure* （配置）菜单中选择 *Select Device* （选择器件），将写作 *Configure>Select Device*。

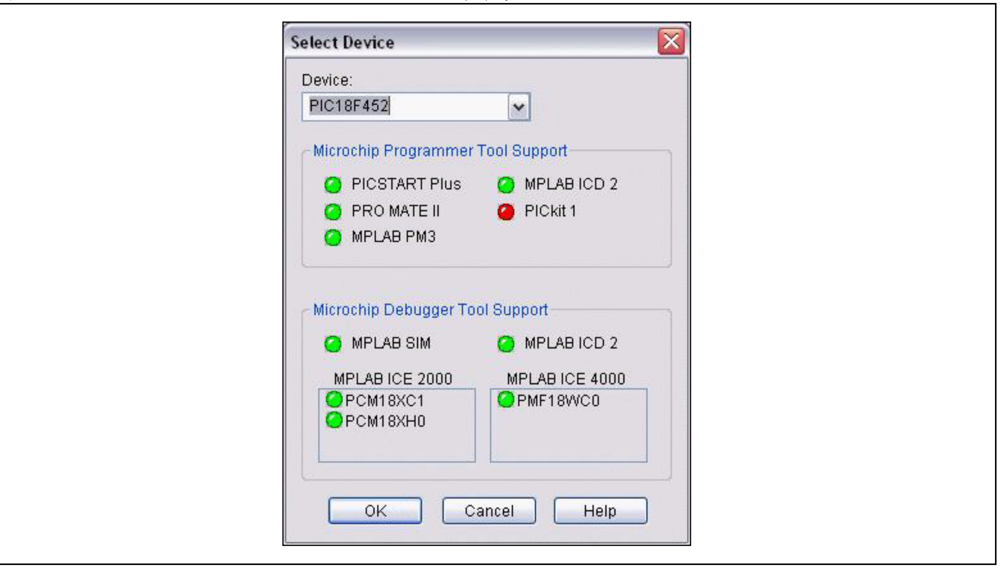
请选择 *Configure>Select Device*。

图 2-2： 选择器件



如果还未选择器件，请从 Device （器件）对话框的下拉列表中选择 **PIC18F452**。

图 2-3： SELECT DEVICE 对话框



“指示灯”表示支持此器件的 MPLAB IDE 组件。

- 绿灯表示完全支持。
- 黄灯表示对即将发布的器件的最小支持，当前版本的某一 MPLAB IDE 组件可能不支持该器件。有黄灯而不是绿灯的组件通常是针对新器件的早期采用者，他们需要这些器件的支持并明白某些操作和功能是不可用的。
- 红灯表示不支持这种器件。将来可能会提供支持或者该工具根本不适合该器件，例如， MPLAB ICE 2000 不支持 dsPIC 器件。

2.5 创建项目

下一步，使用项目向导创建项目。项目是将文件组织起来以便进行编译和汇编的方式。我们将使用一个用于该项目的汇编文件和一个链接描述文件。选择 *Project>Project Wizard*。

在 Welcome （欢迎）对话框中，单击 **Next>** 继续。

下一个对话框 （Step One （第一步））让您选择器件 （已选定）。确定所选择的是 PIC18F452。否则，请在下拉列表中选择 PIC18F452。单击 **Next>**。

图 **2-4**： 项目向导——选择器件



2.6 设置语言工具

Project Wizard 的第二步设置该项目所要使用的语言工具。在 Active Toolsuite （可用工具包）列表框中选择 “Microchip MPASM Toolsuite”（Microchip MPASM 工具包）。然后应可在 Toolsuite Contents （工具包内容）框中看到 “MPASM” 和 “MPLINK”。可以单击任一个查看其位置。如果将 MPLAB IDE 安装到了默认目录，则 MPASM 汇编器的可执行文件位于：

```
C:\Program Files\MPLAB IDE\MCHIP_Tools\mpasmwin.exe
```

MPLINK 链接器的可执行文件位于：

```
C:\Program Files\MPLAB IDE\MCHIP_Tools\mplink.exe
```

以及 MPLIB 库管理器的可执行文件位于：

```
C:\Program Files\MPLAB IDE\MCHIP_Tools\mplib.exe
```

如果未正确显示，使用 Browse （浏览）按钮将它们设置为 MPLAB IDE 子文件夹中的相应文件。

图 2-5： 项目向导——选择语言工具

完成后，单击 **Next>**。

2.7 为项目命名

向导的第三步让您为项目命名并将它存入文件夹。这个示例项目将被命名为 MyProject。使用 Browse 按钮，将项目放在名为 Projects32 的文件夹中。单击 **Next>**。

图 2-6： 项目向导——为项目命名



2.8 将文件添加到项目中

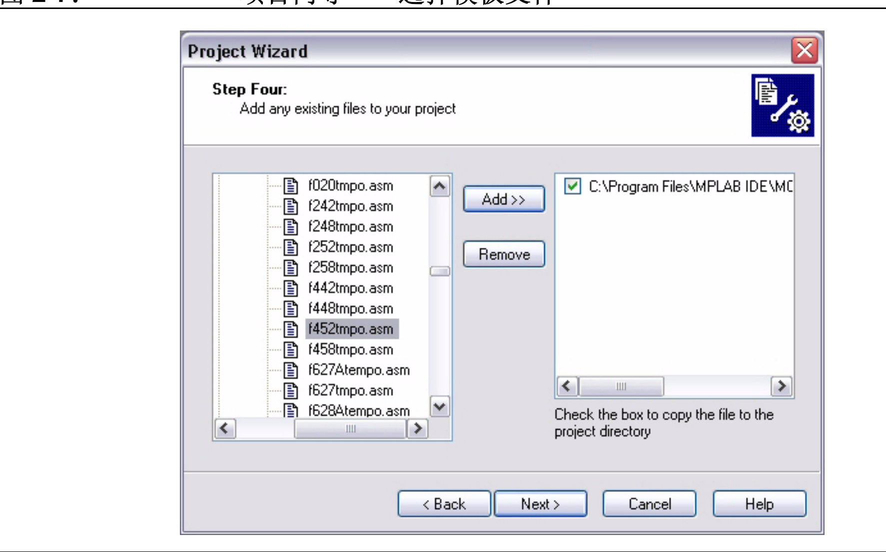
Project Wizard 的第四步允许为项目选择文件。由于还未选择源文件，我们将使用 MPLAB IDE 模板文件。这些模板文件是用于开始项目的简单文件。它们具有任何源文件的基本部分并且包含可帮助您编写和组织代码的信息。这些文件在 `MPLAB IDE` 文件夹中，默认情况下该文件夹位于个人计算机的 `Program Files` 文件夹中。每个 Microchip PICmicro 和 dsPIC 器件都有一个模板文件。

选择名为 `f452tmpo.asm` 的文件。如果将 MPLAB IDE 安装在了默认位置，则该文件的完整路径为：

```
C:\Program Files\MPLAB IDE\MCHIP_Tools\TEMPLATE\Object\f452tmpo.asm
```



图 2-7： 项目向导——选择模板文件



单击 **Add>>** （添加）将文件名移到右边的窗格中，然后单击文件名前面的复选框将此文件复制到项目目录。

接下来，添加项目的第二个文件：链接描述文件。每个器件都有一个链接描述文件。这些文件定义各个器件的存储器配置和寄存器名称。链接描述文件在 `MCHIP_Tools` 文件夹下的 `LKR` 子文件夹中。本项目使用名为 `18F452.lkr` 的文件。该文件的完整路径如下：

```
C:\Program Files\MPLAB IDE\MCHIP_Tools\LKR\18F452.lkr
```

> **注：** 还有一个名为 `18F452i.lkr` 的链接描述文件，在使用 MPLAB ICD 2 （所以名称中有 “i”）时，用于此器件。该链接描述文件为 MPLAB ICD 2 保留了存储区。由于本例将使用软件模拟器，因此不需要使用该链接描述文件。

要将链接描述文件复制到项目中，请单击对应的复选框。

图 **2-8**： 项目向导——选择链接描述文件

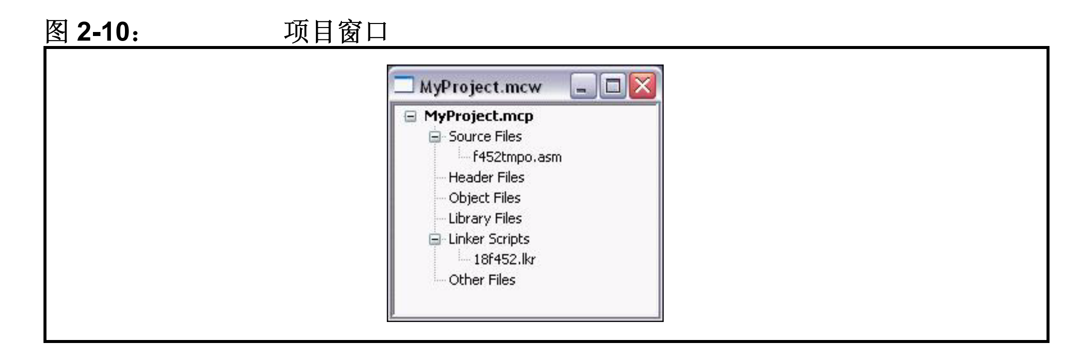
确保您的对话框类似于上图，选中这两个复选框，然后单击 **Next>** 完成项目向导。
项目向导的结束屏幕是一个摘要，显示了所选择的器件、工具包和新项目文件名。

图 **2-9**： 项目向导——摘要



在单击了 **Finish** （完成）按钮之后，复查 MPLAB IDE 界面上的项目窗口。它看起来应该如图 2-10 所示。如果此项目窗口未打开，请选择 *View>Project*。

图 2-10：　　项目窗口



> **提示：**可以通过在项目窗口中使用鼠标右键来添加文件和保存项目。发生错误时，可以通过选择文件并使用单击鼠标右键出现的菜单来手动删除它们。

2.9 编译项目

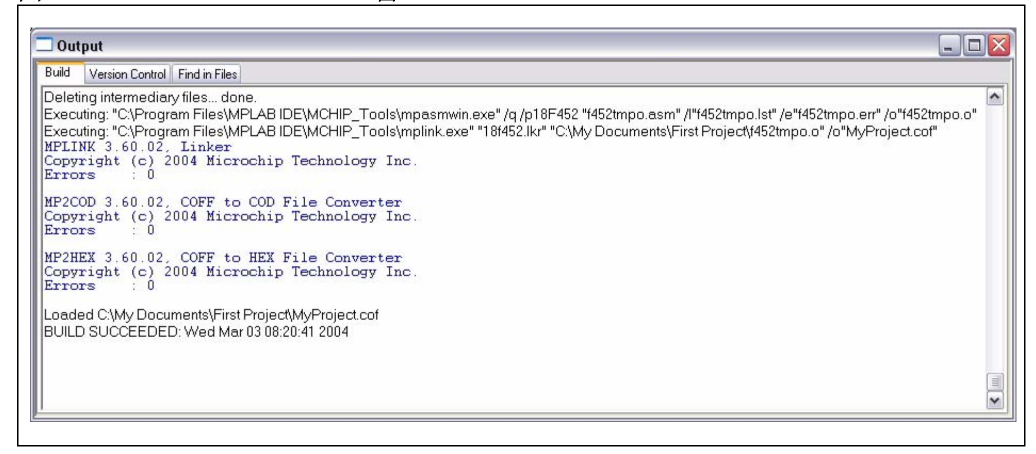
使用 Project （项目）菜单可以汇编和链接当前文件。虽然这些文件中还没有任何代码，但确保了项目是正确设置的。

要编译项目，可选择以下任一种方式：

- *Project>Build All*
- 在项目窗口中右击项目名称并选择 Build All （编译所有）
- 在项目工具栏中单击 Build All 图标。将鼠标悬停在图标上即可看到弹出的文本，显示该图标的含义。

Output （输出）窗口将显示编译的结果。在任何一步中，应该都没有错误。

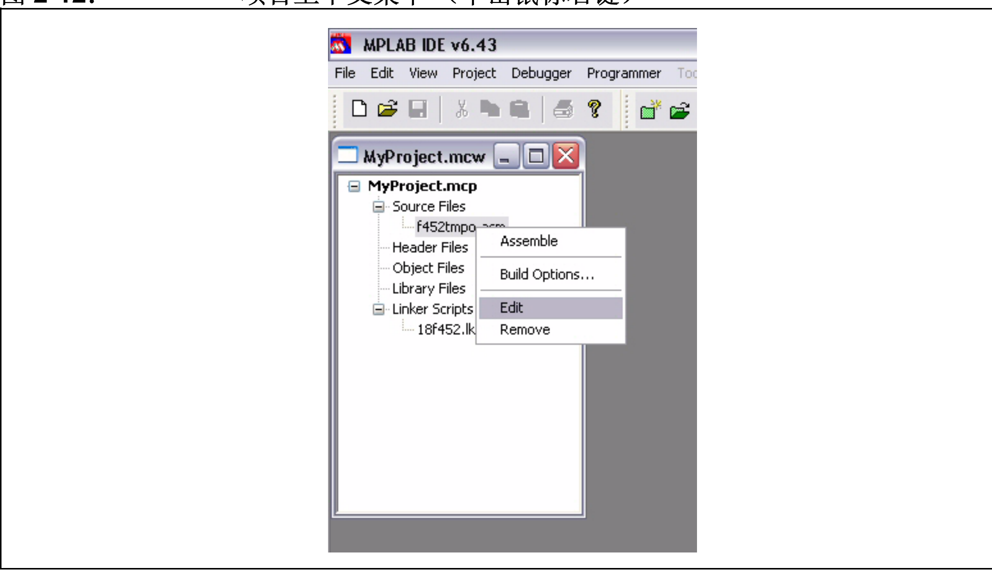
图 2-11：　　OUTPUT 窗口



2.10 创建代码

通过在项目窗口中双击模板文件名，或者通过用光标选择模板文件并用鼠标右键打开上下文菜单，可打开项目中的模板文件：

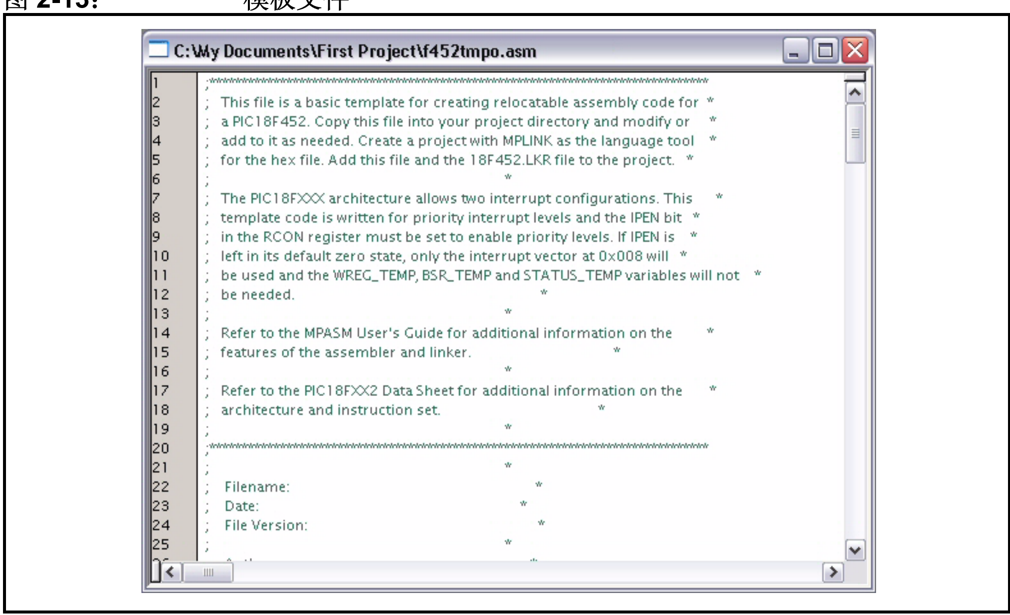
图 2-12： 项目上下文菜单（单击鼠标右键）



在此文件的开头有一些注释，并且此区域可被用作文件的标准注释信息头。现在您可以不管它，但如果这是一个实际的项目，您可以在这里输入与设计有关的信息。

注：	此处显示行数。可以通过右击编辑器窗口，选择 Properties （属性），然后在 Editor Options （编辑器选项）对话框的 **Editor** （编辑器）选项卡中选中 / 清除 “Line Numbers”（行数）复选框，来显示或隐藏行数。

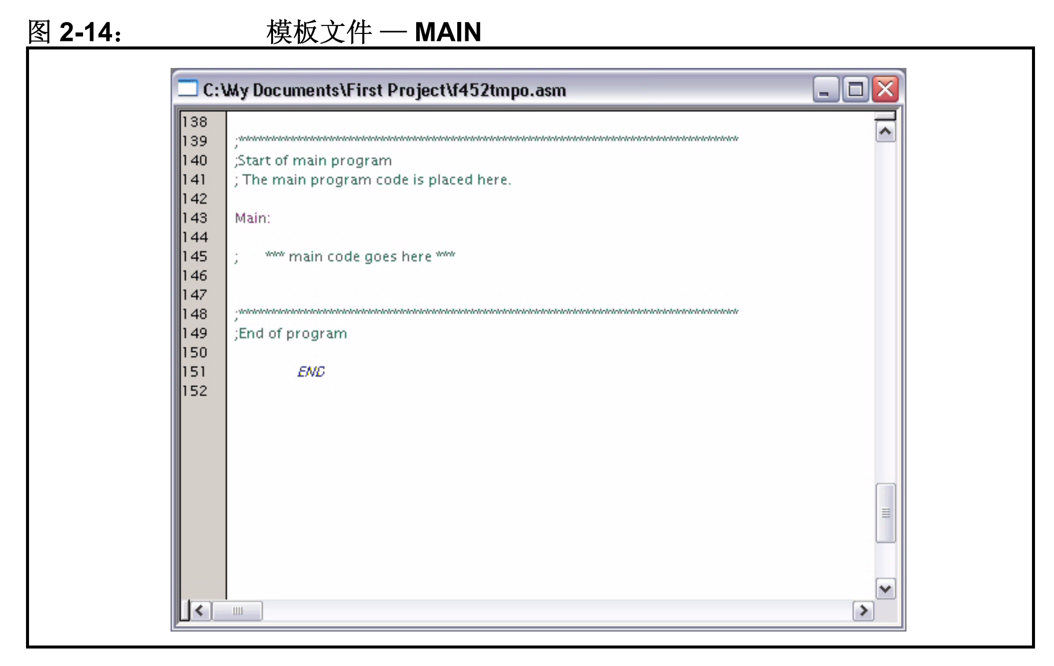
图 2-13： 模板文件



向下滚动到文件底部。

文件第一部分的代码用于较高级的功能，如在最终应用程序中设置中断和配置 位。现在可以忽略这些细节而只专注于编写代码。新代码将放在文件中符号 Main 之后。

图 2-14： 模板文件 — MAIN



打开任何源文件就可以自动进入编辑器。输入以下代码：

```
Main:

    clrf   WREG
    movwf  PORTC; clear PORTC
    movwf  TRISC; configure PORTC as all outputs

Init
    clrf   COUNT
IncCount
    incf   COUNT
    movf   COUNT,W
    movwf  PORTC; display COUNT on PORTC

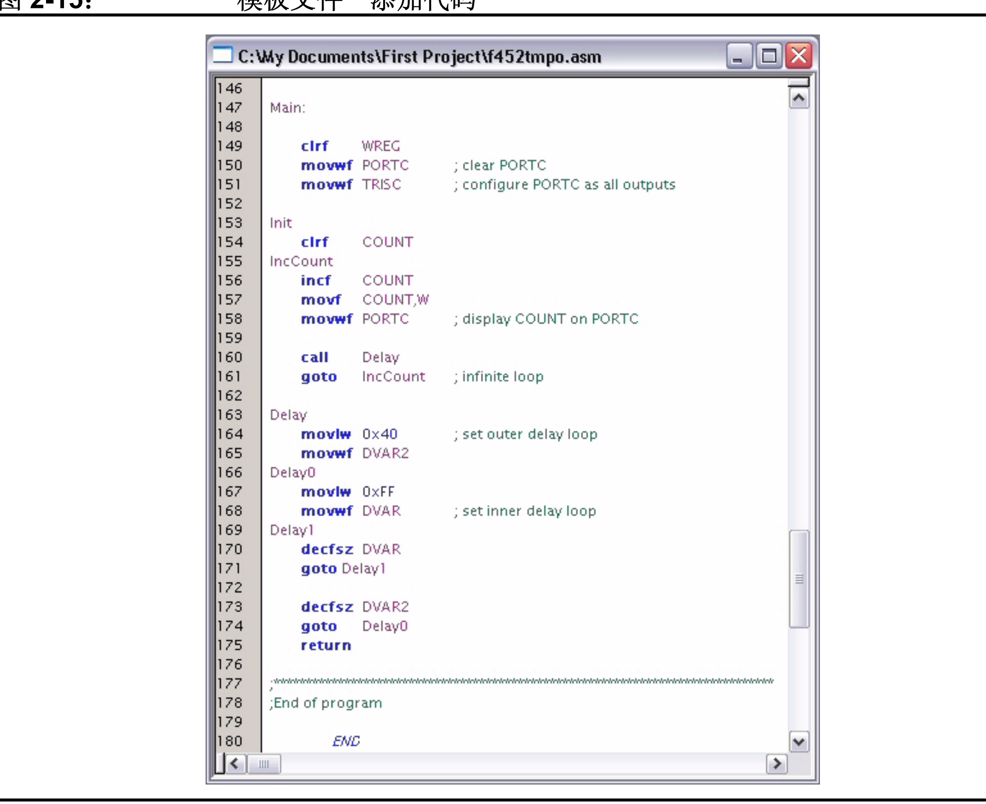
    callDelay
    goto   IncCount; infinite loop

Delay
    movlw  0x40; set outer delay loop
    movwf  DVAR2
Delay0
    movlw  0xFF
    movwf  DVAR; set inner delay loop
Delay1
    decfsz DVAR
    goto   Delay1

    decfsz DVAR2
    goto   Delay0
    return
```

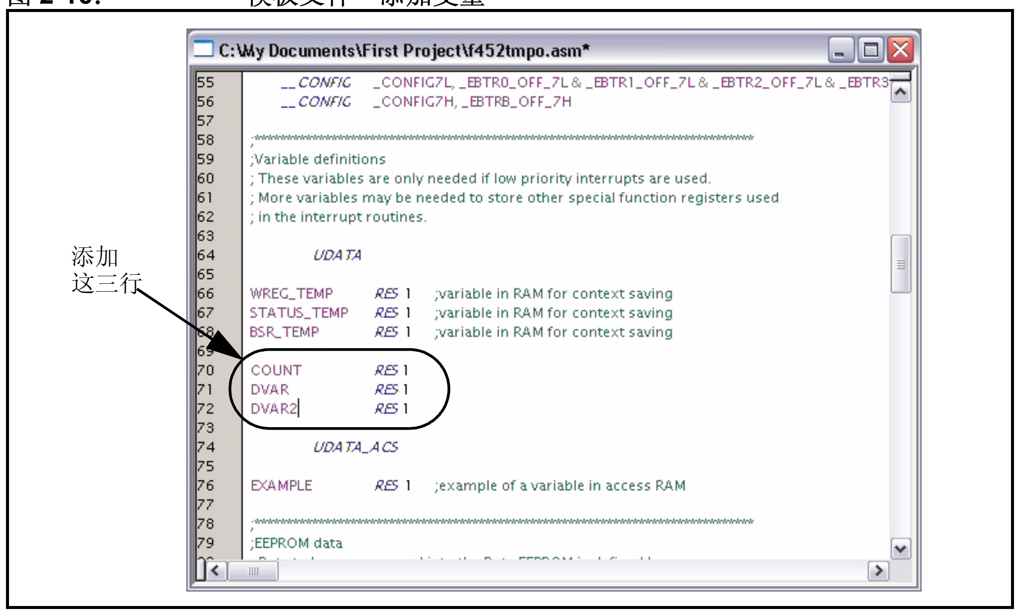
现在模板文件应如图 2-15 所示。

图 2-15：　　模板文件—添加代码



在这个简短的代码中，我们使用了三个变量，分别为 COUNT、 DVAR 和 DVAR2。需要在模板文件中名为 UDATA 的未初始化数据段中定义这些变量。在模板文件的该段中已有三个变量，因此使用相同的格式把我们的变量加在后面即可。每个变量都是 8 位的，因此，只需要为每个变量保留 1 个字节。

图 2-16：　　模板文件—添加变量



2.11 再次编译项目

选择 *Project>Build All* 汇编和链接代码。如果代码汇编的过程没有出错， Output 窗口就如图 2-17 所示：

图 2-17： 编译完成后的 OUTPUT 窗口



如果汇编和链接不成功，检查以下各项然后再次编译项目：

- 检查在编辑器窗口中输入的代码的拼写和格式。确认新变量和特殊功能寄存器 TRISC 和 PORTC 是大写的。如果汇编器在 Output 窗口中报错，双击错误， MPLAB IDE 就会打开源代码窗口，并在左边空白处用绿色箭头指向源代码中相应的行。
- 检查是否为 PICmicro 器件使用了正确的汇编器 （MPASM 汇编器）和链接器。选择 *Project>Set Language Tool Locations*。单击 “+”号框以展开 Microchip MPASM Toolsuite 及其 Executables。单击 MPASM Assembler （`mpasmwin.exe`）然后检查显示的位置。如果位置正确，单击 **Cancel**。如果不正确，更改位置然后单击 **OK**。默认搜索路径可能为空。

在成功编译项目后，会加载语言工具生成的输出文件。此文件包含可被编程到 PICmicro MCU 的目标代码和调试信息，便于调试源代码并在 Watch 窗口中以符号的形式查看源变量。

注：	在现实应用中，需要将多个文件编译 / 汇编和链接成最终的可执行应用程序时，项目的真正作用非常明显。项目会跟踪所有这些过程。可为每个文件设置 Bulid Options （编译选项）来访问语言工具的其他功能，如报告输出和编译器优化。

2.12 使用软件模拟器测试代码

要测试代码，就需要用于执行 PICmicro 指令的软件或硬件。调试执行工具，就是一种硬件或软件工具，用于在程序（在本例中，为 cnt452.asm）执行过程中检查代码。硬件工具（如 MPLAB ICE 或 MPLAB ICD 2）可以在实际的器件中执行代码。如果没有硬件调试器，可以使用 MPLAB 软件模拟器来测试代码。本教程使用的就是 MPLAB SIM 软件模拟器。

软件模拟器是在个人计算机上运行的软件程序，它用于*模拟* PICmicro MCU 的指令。它并不是“实时”运行的，因为软件模拟器程序的运行速度取决于计算机的速度、代码的复杂性、操作系统的开销以及正在运行的其他任务的数目。但是，软件模拟器可精确*测量*代码在应用中实时运行时，所需要的时间。

> **注：** 其他调试执行工具还包括 MPLAB ICE 2000、MPLAB ICE 4000 和 MPLAB ICD 2。可以选择一些硬件工具来测试应用板上的代码。大部分 MPLAB IDE 调试操作和软件模拟器一样，但是与软件模拟器不同的是，这些工具允许目标 PICmicro MCU 以实际目标应用中的全速运行。

选择软件模拟器作为调试执行工具。这可以通过 *Debugger>Select Tool* 下拉菜单完成。在选择了 MPLAB SIM 之后，应可看到以下变化（参见图 2-18 中的相应编号）。

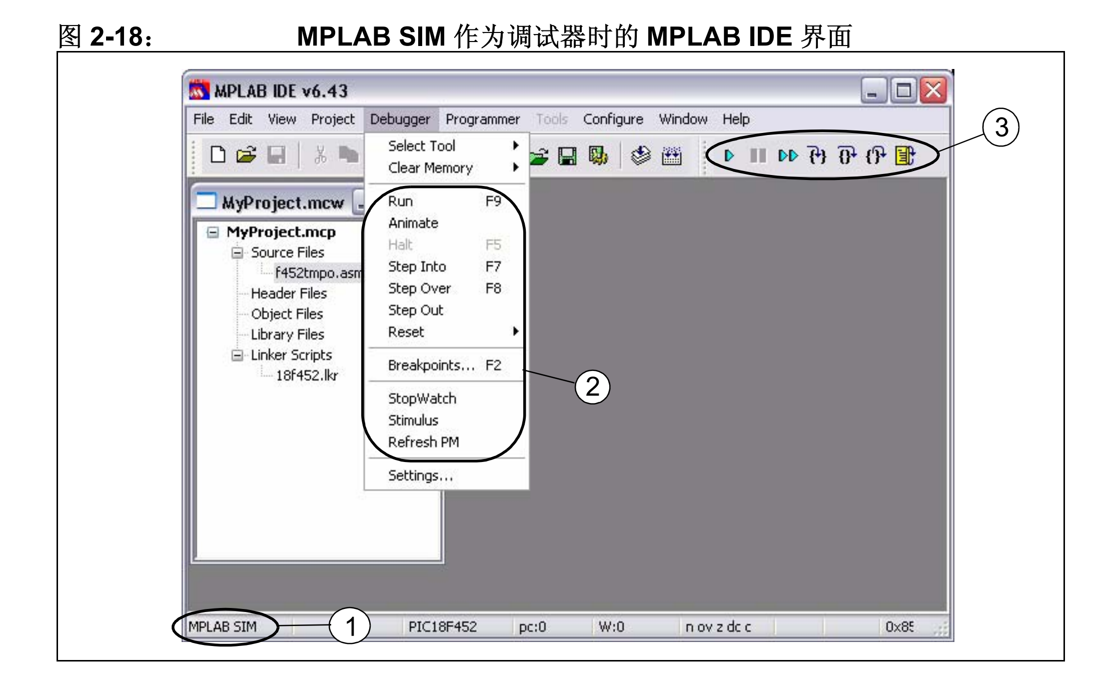
① MPLAB IDE 窗口底部的状态栏应该变为“MPLAB SIM”。

② 现在 Debugger（调试器）菜单中应该会出现更多菜单项。

③ 调试工具栏中应该会出现更多工具栏图标。

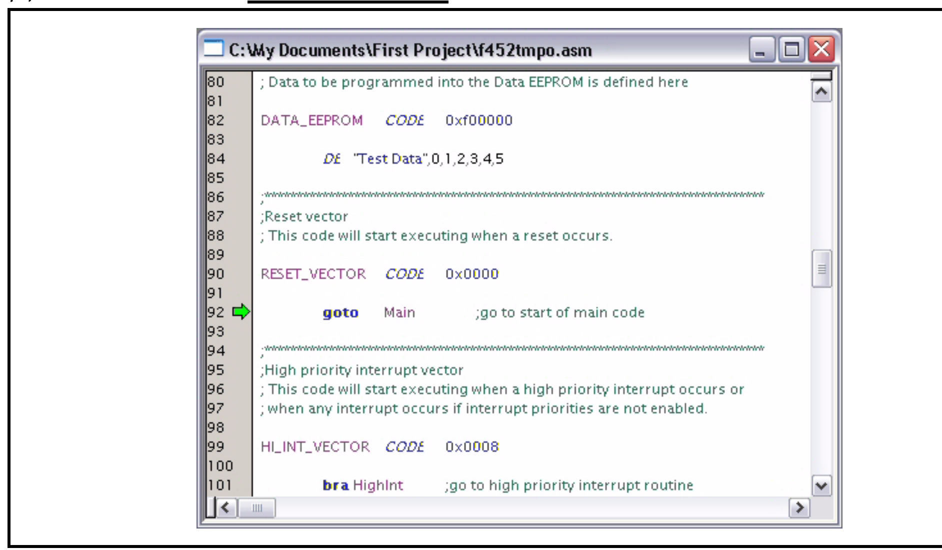
> **提示：**若将鼠标指针悬停于工具栏按钮之上，即可看到有关该按钮功能的简单描述。

图 2-18： MPLAB SIM 作为调试器时的 MPLAB IDE 界面



接下来，选择 *Debugger>Reset*，就会在程序开始之处显示一个绿色箭头。这是模板文件的一部分。此时存储器中的第一条指令将跳至名为 Main 的标号处（您已在此处插入了代码）。该指令跳过存储器中地址较低的 PIC18XXXX 向量区。

图 2-19： *DEBUG>RESET*



要单步执行应用程序，请选择 *Debugger>Step Into*。这将执行当前指向的代码行，然后箭头将指向要执行的下一行代码。

这些常用功能在调试工具栏中也有快捷方式。

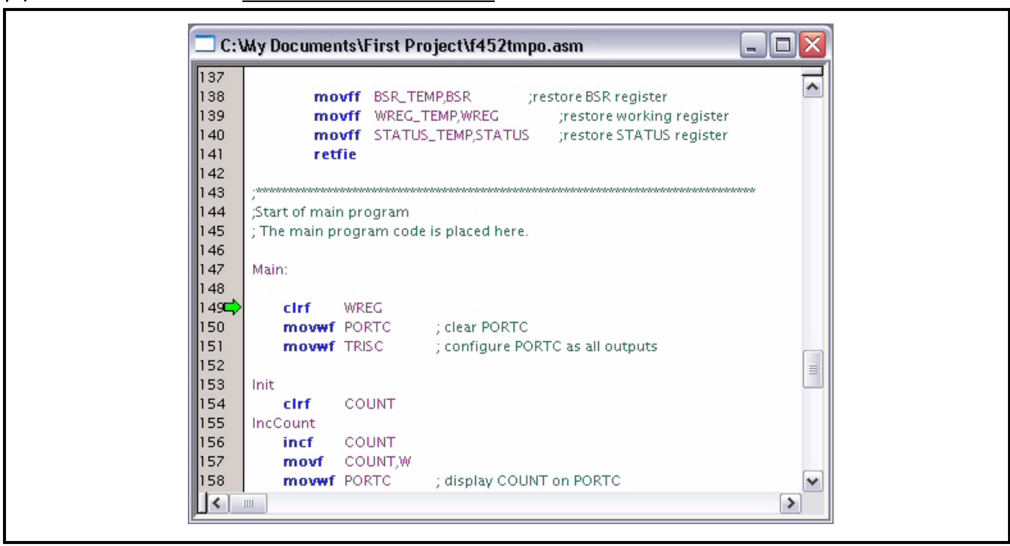
表 2-1： 调试快捷方式图标

Debugger 菜单	工具栏按钮	热键
Run （运行）		F9
Halt （暂停）		F5
Animate （单步连续运行）		
Step Into （单步运行）		F7
Step Over （单步跳过）		F8
Step Out Of （单步跳出）		
Reset （复位）		F6

> **提示：**单击工具栏上相应的图标或者使用菜单项旁边显示的热键。这通常是重复操作的最佳方法。

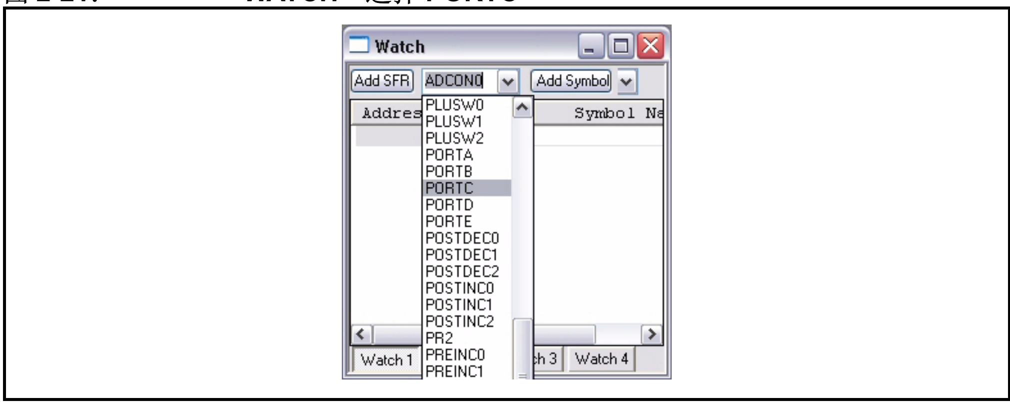
接下来，单击 Step Into 图标或选择 *Debugger>Step Into* 来单步运行 `Main` 中的代码。

图 2-20： ***DEBUG>STEP INTO***



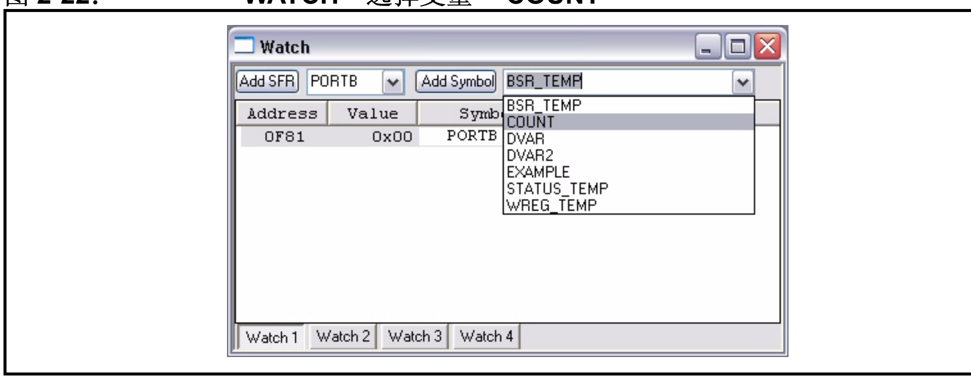
为了查看代码是否按照预期运行，可以向 PORTC 发送递增值，然后观察发送到 PORTC 的值。选择 *View>Watch* 打开一个空的 Watch 窗口。在 Watch 窗口的顶部有两个下拉列表。左边标注为 “Add SFR” 的下拉列表可用于将特殊功能寄存器 PORTC 添加到 Watch 窗口。从列表中选择 `PORTC`，然后单击 **Add SFR** （添加特殊功能寄存器）将它添加到窗口。

图 2-21： **WATCH—选择 PORTC**



右边的下拉列表允许添加程序中的变量。使用此下拉列表将 COUNT 变量添加到 Watch 窗口。从列表中选择 COUNT，然后单击 **Add Symbol** （添加变量）将它添加到窗口。

图 2-22： WATCH—选择变量 “COUNT”



Watch 窗口现在应显示两个寄存器的地址、值和名称。在程序执行的当前时刻，它们均应为零。

注：	也可以从 SFR、 File Register （文件寄存器）或编辑器窗口中将这些项拖拽到 Watch 窗口，或者直接在窗口中的 symbol name （符号名称）下方单击，然后输入这些项，从而将它们添加到 Watch 窗口。

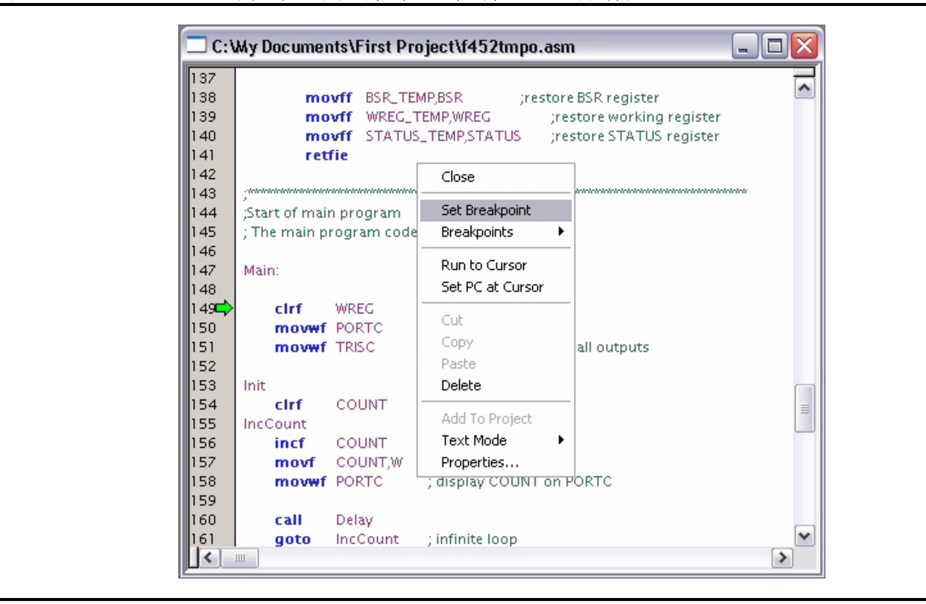
图 2-23： WATCH—复位值





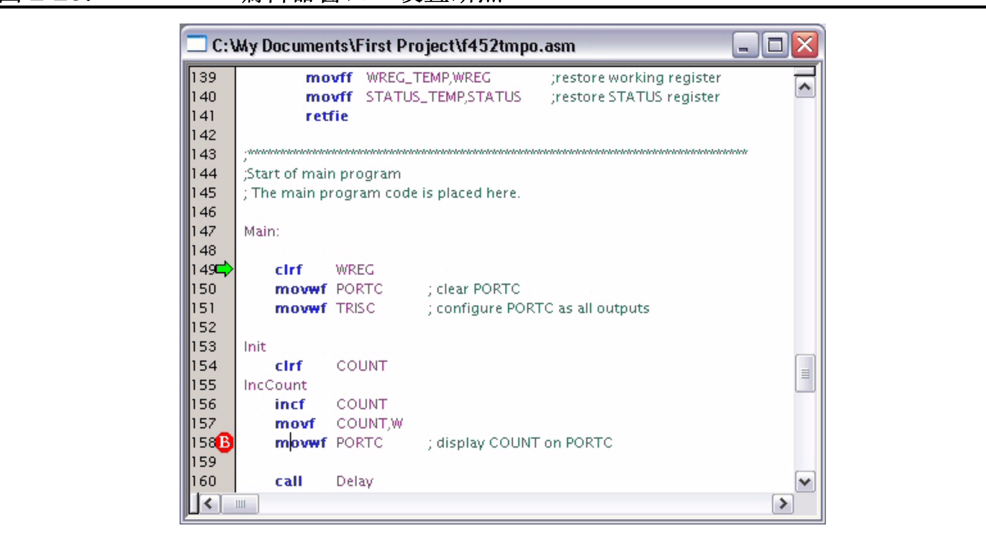
可以继续单步执行代码，但是请将断点设置在向 PORTC 发出第一个值之前。要设置断点，请将光标放在要设置断点的行上，然后单击鼠标右键。

图 2-24：　　　调试上下文菜单（在行上右击鼠标）



从上下文菜单中选择 Set Breakpoint（设置断点）。该行上会出现一个红色的 “B”。（也可以在行上双击鼠标来添加断点。）

图 2-25：　　　编辑器窗口—设置断点



选择 *Debugger>Run* 运行应用程序。在程序暂停在第一个断点之前，状态栏上会短暂地显示文本消息 “Running…”（运行 …）。

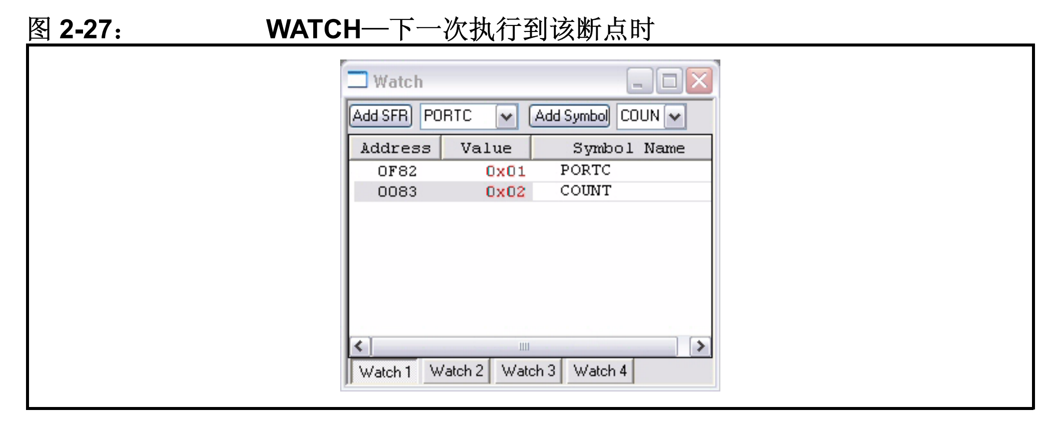
现在 Watch 窗口应显示变量 COUNT 的值已被加 1，但是由于断点位于将值传给 PORTC 的代码执行前的那一行上，因此 PORTC 的值仍为零。

图 2-26： WATCH—执行到断点时



单击 Run 图标执行代码，直到再次到达该断点。现在 Watch 窗口应显示两个值均被加 1。

图 2-27： WATCH—下一次执行到该断点时

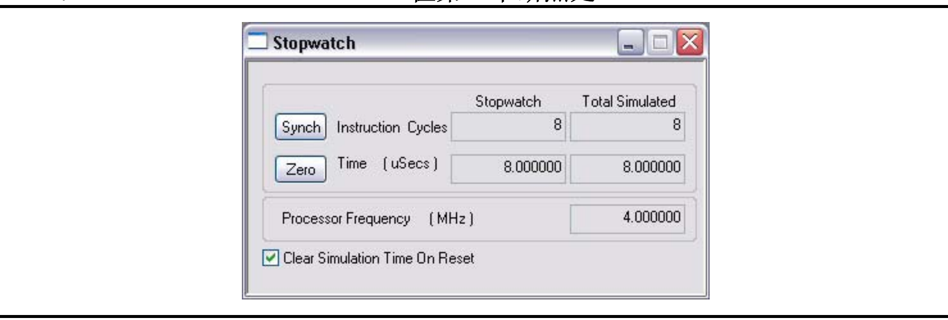


这似乎表示程序按照设计正常运行的。您可以单步运行代码或者多次运行代码以确认它是否正确执行。如果单步运行进入延时循环，您将需要不断执行几千次单步运行，才能结束循环。要跳出这种延时循环，可以使用 *Debugger>Step Out*。

如果对计算延时时间感兴趣，可以使用数据手册确定运行延时循环中的每个指令所需要的时间并由此得出一个相当精确的数字。也可以使用 MPLAB StopWatch （跑表）来测量延时时间。重点应该关注显示 COUNT 的每个新值时的时间。如果像最初那样在从 COUNT 移动到 PORTC 的指令上设置断点，那么就可以运行程序到同一位置的下一个断点来测量该时间。

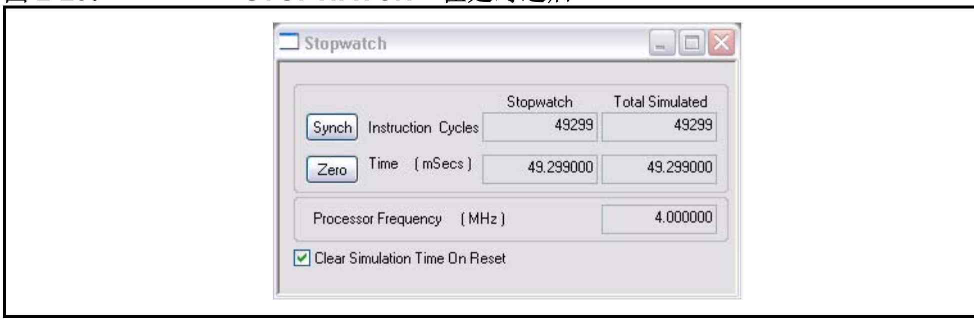
使用 *Debugger>StopWatch* 打开 StopWatch 对话框。确定在 `movwf COUNT` 指令处设置了断点，然后依次单击 *Debug>Reset* 和 *Debug>Run*，程序将在 `movwf COUNT` 指令处暂停。若采用默认的 4 MHz 的处理器频率， StopWatch 将显示到达第一个断点花了 8 us。

图 2-28： STOPWATCH—在第一个断点处



再次执行 Run 来运行一次循环， StopWatch 显示它花费了 49ms。要更改此时间，可以改变延时循环中的值。

图 2-29： STOPWATCH—在延时之后



2.13 教程总结

通过学习本教程，您已经执行了创建、编译和测试简单项目的主要步骤。完成的任务包括：

选择器件，即 PIC18F452。

使用 Project Wizard 创建项目，并使用该向导完成以下任务：

- 选择 MPLAB IDE 中内置的 MPASM 汇编器和 MPLINK 链接器语言工具；
- 为项目添加文件：用于选定器件的模板文件和用于正确编译项目的链接描述文件。

编写一些简单的代码，将变化的值发往 I/O 端口。

编译项目。

最后，使用软件模拟器测试代码。

这些是开始使用 MPLAB IDE 的基本步骤。现在您已准备就绪，可继续探索 MPLAB IDE 的功能了。

注：



第 3 章 步骤和教程

3.1 简介

此步骤和教程是专为帮助初学者和有经验的应用开发人员而设计的。本章给出了对使用 MPLAB IDE 和各种 Microchip 开发工具进行开发的逐步说明，其中，每一步都首先讨论了通用信息，然后提供具体的使用示例。将每步的示例连起来就构成一个应用示例。

入门

- 选择器件
- 设置配置位
- 使用编辑器创建源代码

创建、设置和编译项目

- 创建新项目
- 使用项目向导
- 设置语言工具包
- 为新项目命名和定位
- 添加文件
- 完成项目
- 查看项目窗口
- 设置编译选项
- 编译项目

调试

- 选择调试器
- 运行代码
- 查看 Debug 窗口
- 使用 Watch 窗口
- 使用断点

编程

- 选择编程器
- 对器件进行编程

其他资源

- 使用 Microchip 帮助

3.2 选择器件

要开始开发应用程序，请选择想要为其开发代码的 Microchip 器件。

要选择器件：

1. 选择 *Configure>Select Device*。
2. 在 Select Device 对话框中，通过 Device 列表框选择器件。 Microchip Tool Support （Microchip 工具支持）部分将显示支持该器件的工具。
3. 单击 **OK** （确定）。

此时状态栏中将会显示所选择的器件。

教程步骤 1：

选择 PIC18F8720 单片机作为要使用的器件。许多 Microchip 工具均支持此器件。

3.3 设置配置位

可以通过对配置位编程 （读作 0）或不编程 （读作 1）来选择不同的器件配置。欲知更多信息，请查询器件的数据手册。

3.3.1 在 MPLAB IDE 或在代码中设置配置位

可以在代码中设置配置位 （即，使用 `__config` 命令或初始化代码）；也可以使用 Configuration Bits （配置位）窗口 （*Configure>Configuration Bits*）设置配置位。虽然可以使用两种方法，但是也许应该仅选择一种方法以避免混淆和改写位值。

如果想要在窗口中更改配置位的值，在开发完成前请不要在代码中更改配置位。而是等到开发完成以后才将窗口值加入代码，重新编译项目 （该过程会将代码值输入窗口）并执行最终运行以确保一切正确。

如果想要在代码中更改配置位的值，请不要对窗口进行任何更改。每当编译项目时，都会将代码值装入窗口，此时先前对窗口所做的任何更改都将丢失。

3.3.2 保存 / 清零配置位的值

当关闭 / 打开工作区时将保存 / 恢复 Configuration bit （配置位）窗口的值。

如果在 Settings （设置）对话框 （*Configure>Settings*）的 **Program Loading** （程序装载）选项卡中选择了 “Clear program memory upon loading a program”（在装载程序时清空程序存储器）复选框，配置位就会在装载程序 （即编译、导入或打开项目）时被清零。

3.3.3 使用外部存储器时的设置

如果器件支持外部存储器，并且在配置位中选择了使用外部存储器的模式，就还需要在 External Memory Settings （外部存储器设置）对话框 （*Configure>External Memory*）中设置所使用的外部存储器的大小。

教程步骤 2：

对于所选择的器件：

- 将使用默认的配置位值，因此不会对 Configuration Bits 窗口进行任何更改。
- 检查 *Configure>Settings* 的 **Program Loading** 选项卡中的设置。应该选中："Clear memory before building a project"（在编译项目之前清空存储器）和 "Clear program memory upon loading a program"（在装载程序时清空程序存储器）。
- PIC18F8720 器件可以使用外部存储器。但在本教程将使用单片机模式或者所有的片上存储器。

3.4 使用编辑器创建源代码

选择 *File>New* 打开一个空的编辑器窗口，可以在该窗口中输入源代码。要保存您的工作，请选择 *File>Save*。

欲知更多有关编辑器的信息，请参阅**第 11 章 "使用编辑器"**。

教程步骤 3：

使用 Windows 资源管理器创建 / 定位存放应用程序项目文件的文件夹 （目录）。然后，在编辑器窗口中输入以下信息，或是将以下信息剪切和粘贴到编辑器窗口中。完成以后，将文件以文件名 `cnt8720.asm` 保存到该项目文件夹下。

```
     title "PIC18F8720 Counting Program"
        #include <p18f8720.inc>

        udata  0x60
COUNT  res 1

RST     code   00h
        goto   Start

PGM     code
Start   clrf   WREG       ;clear W register
        movwf  PORTC      ;clear PORTC
        movwf  TRISC      ;config PORTC as outputs
Init
        clrf   COUNT      ;clr count
IncCount
        incf   COUNT,F    ;increment count
        movf   COUNT,W
        movwf  PORTC      ;display on port c
        goto   IncCount   ;loop

        end
```

3.5 创建新项目

项目包含编译和测试应用程序所需的源文件。项目配置 （工作区）包括以下内容：处理器和显示信息，如所有打开窗口的性质、大小和位置以及工具栏、执行和调试设置的相关信息。欲知更多有关项目和工作区的信息，请参阅**第 4 章 “项目和工作区”**。

可以在 Settings （设置）对话框 （*Configure>Settings*）的 **Project** 和 **Workspace** 选项卡中选择项目和工作区的全局设置。

教程步骤 4：

在本教程中，请保留项目和工作区的默认设置。最明显的就是这意味着您将使用一对一的项目—工作区模型。

3.6 使用项目向导

要在当前工作区中创建新的项目，请使用 Project Wizard。

- 选择 *Project>Project Wizard*。
- 出现 “项目向导——欢迎”屏幕。单击 **Next>** 继续。
- 出现 “项目向导——选择器件”屏幕。验证此处显示的是否是您已经在 Select Device 对话框中选定的器件。要更改器件，请使用下拉菜单。单击 **Next>** 继续。

教程步骤 5：

您将使用步骤 1 中选择的器件，因此不需要进行任何更改。

3.7 设置语言工具包

现在将添加项目中要使用的语言工具。您将看到 “项目向导——选择语言工具包”屏幕。

- 从 “Active Toolsuite”（可用工具包）下拉菜单中选择项目要使用的语言工具包（由 Microchip 或第三方提供）。从该菜单中只可选择已经安装的并且可以和先前选择的器件配合工作的语言工具包。要查看所有现有的 （已安装的）工具包，请选中 “Show all installed toolsuites”（显示所有已安装的工具包）。

  如果在这个列表中仍然找不到所需要的工具包，请单击 **Help! My Toolsuite Isn't Listed!** （需要帮助！没有列出我所需的工具包）以获取更多信息。
- “Toolsuite Contents”（工具包内容）下的框中将显示所选工具包中的工具列表。前面有红 “X”的工具表示该工具未安装或者 MPLAB IDE 无法获得该工具的可执行文件的路径。要指定或检查工具所对应的可执行文件的位置分配，请单击工具，以在 “Location of Selected Tool”（所选工具位置）下显示出可执行文件及其路径。在该文本框中输入或修改路径，或者单击 **Browse** 查找可执行文件。
- 单击 **Next>** 继续。

教程步骤 6：

在本教程中，根据以下步骤设置语言工具：

- 在 “Active Toolsuite”中选择 “Microchip MPASM Toolsuite”（Microchip MPSAM 工具包）。
- 在 “Toolsuite Contents”中逐次单击 MPASM Assembler、 MPLINK Object Linker 和 MPLIB Object Librarian 以确认它们的位置。单击 **Browse** 找到未列出的可执行文件。

此工具包随 MPLAB IDE 一起安装，并且其工具可执行文件位于 MPLAB IDE 安装目录下的 `MCHIP_Tools` 子目录中。

3.8 为新项目命名和定位

现在将输入新项目的名称和路径。您将看到 “项目向导——为项目命名”屏幕。

要设置目录：

- 输入现有目录的路径，或是输入新目录的路径。输入新的路径时，单击 **Next>** 会提示创建目录。
- 单击 **Browse** 找到现有的目录或是想要放置新目录的上一级目录。在 Browse for Folder （查找文件夹）对话框中单击 **OK**。如果正在创建新目录，请在完成路径后单击 **Next>**。如果目录不存在将会提示创建目录。

教程步骤 7：

在 “Project Name”（项目名称）中输入 “cnt8720”。然后单击 **Browse** 并找到在步骤 3 中创建源代码时设置的应用程序项目目录。

3.9 添加文件

现在将向项目中添加文件。您将看到 “项目向导——添加现有文件”屏幕。

- 选择要添加的文件。单击一个文件名以选择该文件。按住 Ctrl 键并单击以选择多个文件。单击 **Add>>** （添加），要添加到新项目中的文件将出现在右面的列表中。
- 要将文件从列表中删除，请单击文件名选择要删除的文件，然后单击 **Remove** （删除）。
- 单击 **Next>** 继续。

教程步骤 8：

将在项目中添加两个文件：

- 找到文件 “cnt8720.asm”并单击它。然后单击 **Add>>** 将它添加到项目中。
- 在 MPLAB IDE 安装目录下的 `MCHIP_Tools/LKR` 子目录中找到 PIC18F8720 器件的链接描述文件 `18F8720.lkr`。

注：	如果计划使用 MPLAB ICD 2 作为调试工具，则请选择 `18F8720i.lkr` 文件。

选中该文件，然后单击 **Add>>** 将它添加到项目中。并且还请选中所添加的文件旁边的复选框以将该文件复制到项目目录。

3.10 完成项目

现在应该检查项目设置。将显示 “项目向导——摘要” 屏幕。

- 检查项目的摘要信息。如果有任何错误信息，请使用 **Back** 返回到需要更改信息的对话框。
- 当对项目设置满意时，单击 **Finish**。

教程步骤 9：

单击 **Finish** 完成项目设置。

3.11 查看项目窗口

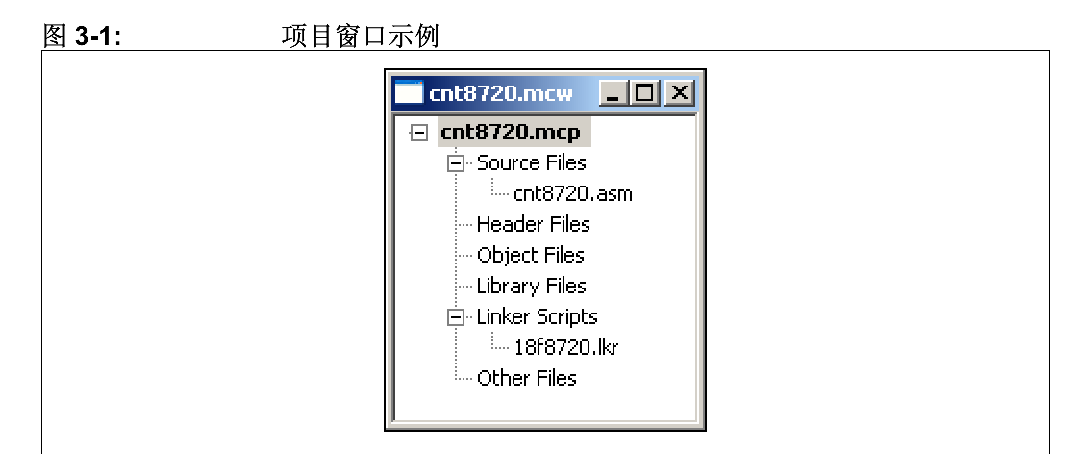
如果项目窗口尚未打开，请选择 *View>Project* 将它打开。工作区的名称将显示在项目窗口顶端的栏中。而项目名称则应该显示在项目窗口内部的最上方。项目名称下还将显示文件类型和任何已添加的文件的树型结构。

教程步骤 10：

项目窗口应该如图 3-1 所示。

- 右击树型结构中的各项以查看相应的弹出菜单。
- 右击 “Source Files”（源文件），将看到 “Add Files”（添加文件）菜单项，这意味着在初始设置之后，还可以将更多的源文件添加到项目中。右击文件 `cnt8720.asm`，将看到 “Remove”菜单项，通过该方式可以将文件从项目中删除。
- 右击窗口中的空白区域并在弹出菜单中选择 “Preferences”（首选项）。从而打开 “Project-Display Preferences”（项目——显示首选项）对话框。
- 虽然本教程不会更改任何首选项，但仍请注意 Version Control （版本控制）部分。欲知更多有关在项目中使用版本控制文件的信息，请参阅**第 4.6 节 “使用版本控制系统 （VCS）”**。
- 单击 **Cancel** 关闭对话框。

图 3-1: 项目窗口示例



3.12 设置编译选项

MPLAB IDE 对项目中所要包含的工具和文件具有默认设置。然而，您可能想要或需要修改这些设置。

要设置编译选项，请选择 *Project>Build Options>Project* 或在项目窗口中右击项目，并在弹出菜单中选择 Build Options （编译选项）。

- 将打开 Build Options 对话框。
- 单击 **General** （常规）选项卡，输入文件的路径或者单击 **Browse** 按钮查找路径，这些文件可能是输出文件、 include 文件、库文件或链接描述文件。

注：	这些是 MPLAB IDE 路径。并不是所有的语言工具都会使用此信息。

- 单击具体的语言工具的选项卡 （如 MPASM Assembler）并设置该工具的操作特性。
- 单击 **OK**。

要改写某个特定项目文件的项目设置，例如 `ProjFile1.asm`，请选择 *Project>Build Options>ProjFile1.asm*，或右击项目窗口中的 `ProjFile1.asm`，并从弹出菜单中选择 Build Options。

- 将打开 File Settings （文件设置）对话框。
- 单击具体的语言工具的选项卡 （如 MPASM Assembler）并设置该工具的操作特性。
- 单击 **OK**。

教程步骤 11：

设置项目编译选项：

- 选择 *Project>Build Options>Project*。
- 单击 **MPLINK Linker** 选项卡。选中 "Generate map file"（生成映射文件）复选框。
- 单击 **OK**。

3.13 编译项目

现在已经准备就绪，可以编译项目了。可以通过以下方式选择 Build All。

- 菜单栏 （*Project>Build All*）
- 弹出菜单 （右击项目窗口中的项目名称）
- 工具栏 （ ）。

在编译的过程中：

- 如果选用了 MPASM 汇编器，就会打开一个 状态窗口，显示编译的进度和最后的结果。编译完成后将关闭该窗口。
- 还将打开 Output 窗口。该窗口将显示有关编译的信息，包括遇到的任何错误。

如果项目编译失败，请检查以下各项，然后重新编译项目：

- 检查在编辑器窗口中输入的所有代码的拼写和格式。如果在 Output 窗口中报告有错，双击该错误就会在源代码窗口左侧的灰色区域中以绿色箭头指出源代码中出错的行。
- 检查用于该项目及其项目文件的语言工具是否正确。

编译成功后，将装载由语言工具生成的调试文件 （*.cod， *.cof， *.elf 等）。该文件允许您调试源代码，并在 Watch 窗口中以符号的形式查看变量。

教程步骤 12：

根据以上说明，通过选择 Build All 编译项目。应该能够成功地编译代码。如果确实遇到了编译错误，请根据以上说明对编译错误进行诊断。

3.14 选择调试器

通过 *Debugger>Select Tool* 选择工具以帮助调试代码。 Microchip 提供以下几种类型的调试工具：

- **MPLAB SIM** 软件模拟器——用软件模拟器件操作。
- **MPLAB ICD 2** 在线调试器——使用 ICD 和带有内置调试电路的专用闪存器件来在器件中调试代码。参见 MPLAB IDE 目录中的 Readme 文件以查看所支持的器件的列表。
- **MPLAB ICE 4000** 在线仿真器——通过访问器件的存储器单元，用硬件仿真具有较大存储容量的 PIC18 MCU 和 dsPIC DSC 的操作。参见 MPLAB IDE 目录中的 Readme （自述）文件以查看受支持的器件的列表。
- **MPLAB ICE 2000** 在线仿真器——通过访问器件的存储器单元，用硬件仿真大多数 PICmicro MCU 的操作。参见 MPLAB IDE 目录中的 Readme 文件以查看所支持的器件的列表。

一旦选定了调试工具，就会发现 IDE 发生了以下改变：

- Debugger 菜单——几个标准的选项将变为可用。还可能会出现其他工具特定的选项。该菜单的大部分选项均会出现在源代码 （文件）窗口或 Program Memory 窗口的右击弹出菜单中。
- View （查看）菜单——根据选定的调试器，还可能会出现其他调试选项。
- 调试工具栏——几个标准的选项将变为可用。还可能会出现其他工具特定的选项。
- 状态栏——现在应该在此显示出选定的调试工具。

教程步骤 13：

如果已经购买了用于开发应用程序的硬件或第三方工具，现在请在 Debugger 菜单中选中它。如果它在菜单中不可视 （灰掉了），请查询工具自带的文档，以在 MPLAB IDE 中正确安装和设置该工具。

如果没有其他工具，请选择 MPLAB IDE 中内置的软件模拟器 MPLAB SIM。

3.15 运行代码

不管选择了哪种调试工具，都需要运行 （执行）代码以对代码进行调试。

- 选择 *Debugger>Reset>Processor Reset*。将在源代码窗口左侧的灰色区域中显示一个实心的绿色箭头，表明这是要执行的第一行源代码。
- 选择 *Debugger>Run* 运行应用程序。在状态栏中将显示 “Running…”，并且灰色区域中的箭头将变为空心的。
- 要暂停程序执行，请选择 *Debugger>Halt*。实心的绿色箭头将指向应用程序停止时所在的代码行。
- 也可以通过选择 *Debugger>Step Into* 单步运行应用程序。这将执行当前代码行，然后箭头将指向要执行的下一行代码。

注：	也可以单击工具栏上相应的图标，或者使用菜单项旁边显示的热键而不是选择菜单项来选择相应的调试命令。这是在调试时重复地运行、暂停和单步执行的最佳方法。

教程步骤 14：

复位代码 （*Debugger>Reset*），并通过单击工具栏上的 Run 图标运行代码。（将鼠标悬停在工具栏图标上可查看图标的含义。）然后通过单击 Halt 图标停止程序。

3.16 查看 DEBUG 窗口

可以在标准的调试窗口中查看程序存储器、数据存储器以及其他类型的器件存储器的内容，以辅助调试应用程序。这些窗口在 View 菜单上总是可见的。但是，若器件或调试工具不支持某些项，这些项可能就会灰掉。欲知更多有关每个调试窗口的信息，请参阅**第 8 章 “MPLAB IDE 窗口”**。

- Disassembly Listing 窗口
- Hardware Stack 窗口
- Program Memory 窗口
- File Registers 窗口
- EEPROM 窗口
- LCD Pixel 窗口
- Watch 窗口
- Special Function Registers 窗口

菜单中可能还会出现其他调试窗口，这取决于选定的调试工具。

- Trace Memory 窗口 （对于模拟器和仿真器）

教程步骤 15：

可能会使用以下窗口观察正在运行的代码：

- 文件 （编辑器）窗口——显示实际的汇编或 C 代码。
- Program Memory 窗口——以十六进制操作码 （Opcode Hex）、机器码或符号形式显示代码。
- Disassembly Listing 窗口——显示反汇编后的代码。
- Watch 窗口——显示指定要 “观察”的寄存器和位的值。

应该已经打开了包含代码的文件窗口。在下一步中将使用 Watch 窗口。

3.17 使用 WATCH 窗口

要查看应用程序的执行过程，就需要打开调试窗口来观察寄存器的值。虽然 File Register 窗口和 Special Function Register 窗口可以显示许多寄存器的值，但是通常只希望查看少量的特定寄存器。要设置将要查看的寄存器的列表，请使用 Watch 窗口（*View>Watch*）。

要将项目添加到 Watch 窗口中：

- 请在下拉列表中选中某个 SFR 或变量，然后单击相应的 Add 按钮。
- 将 SFR 窗口、 File Register 窗口或 File 窗口中的某项拖到 Watch 窗口中。
- 直接在窗口中的 symbol name 下方单击，以输入 SFR 或变量名。

首先列出的是符号的文件寄存器地址，接下来是符号名称，最后是符号的值。

欲知更多有关 Watch （观察）窗口的信息，请参阅**第 8.12 节 “Watch 窗口”**。

教程步骤 16：

首先设置 Watch 窗口：

- 在窗口顶部的变量选择框中选择 COUNT。单击 **Add Symbol** 将它添加到 Watch 窗口列表中。若想在下拉列表中快速查找该变量，请在选中下拉图标后输入 COUNT。
- 在窗口顶部的 SFR 选择框中选择 WREG。单击 **Add SFR** 将它添加到 Watch 窗口列表中。若想在下拉列表中快速查找 WREG，请在选中下拉图标后输入 WREG。
- 在窗口顶部的 SFR 选择框中选择 PORTC。单击 **Add SFR** 将它添加到 Watch 窗口列表中。若想在下拉列表中快速查找 PORTC，请在选中下拉图标后输入 PORTC。

现在应该已经将这些符号添加到了 Watch 窗口中了。接下来，将运行代码，并在单步执行程序的过程中观察变量和 SFR 值的变化。

- 选择 *Debugger>Reset>Processor Reset* 复位应用程序。
- 选择 *Debugger>Step Into* （或者单击等效的工具栏图标）直到代码运行至以下程序行：

```
incf COUNT,F ;increment count
```

- 再执行一步，将在 Watch 窗口中看到 COUNT 的值从 00 变为了 01。
- 接着再执行一步，将在 Watch 窗口中看到 WREG 的值从 00 变为了 01。
- 然后再执行一步，将在 Watch 窗口中看到 PORTC 的值从 00 变为了 01。

注：	若在前面的调试操作中对 Watch 窗口中的值做了更改，相应值就会呈现彩色，若未更改则为黑色。

3.18 使用断点

您可以使用断点来使程序有条件地暂停，这样就可以在运行一段时间之后观察存储器、寄存器或变量的值。可以在文件 （编辑器）窗口、程序存储器窗口或反汇编窗口设置断点。

3.18.1 断点设置

可采用以下方法设置断点：

1. 双击灰色区域——双击想要设置断点的那行代码旁边的灰色区域。再次双击可删除断点。
2. 弹出菜单——将指针放在想要设置断点的那行代码上。然后，右击鼠标，在弹出菜单中选择 Set Breakpoint （设置断点）。一旦设置了断点， Set Breakpoint 就会变为 Remove Breakpoint （删除断点）和 Disable breakpoint （禁止断点）。弹出菜单中 Breakpoints 下的其他选项用于删除、启用或禁止所有断点。
3. **Breakpoint 对话框**—— 打开 Breakpoint 对话框 （*Debugger>Breakpoints*）以设置、删除、启用或禁止断点。必须在选择调试工具后才可以使用该选项。

3.18.2 断点符号

在每个支持的窗口左侧的灰色区域中均将显示与断点操作相关的符号。欲知有关这些符号的信息，请参阅**第 8.3 节 “在代码窗口中显示符号”**。

3.18.3 代码中的断点

更改源代码时将会自动调整断点。这意味着：

- 在地址上设置的断点将会始终位于这些地址，与该地址的代码无关。
- 在源代码行中设置的断点将会随着这些代码行移动。

当在 C 代码中设置断点时，若调试时需要在 C 代码列表和相关的汇编列表中切换，必须十分小心。

- 当在一行 C 代码上设置断点时，在汇编代码中的对应的第一行代码上也会同时设置断点。在两个列表相应的代码行边均会显示一个红色的断点符号。
- 当在汇编列表中对表示一行 C 代码的任一行汇编代码设置断点时，汇编代码行旁会出现一个红色的断点符号，而 C 代码行旁会出现一个黄色的断点符号。
- 当在 Program Memory 窗口中删除对表示一行 C 代码的任一行汇编代码所设置的断点时，汇编代码行旁红色的断点符号将消失，而 C 代码行旁仍会保留黄色的断点符号，直到删除了所有对应的汇编代码断点为止。

3.18.4 断点和 MPLAB ICD 2

MPLAB ICD 2 在线调试器所支持的断点数是有限的。可用的数目显示在 Breakpoint 对话框中 （*Debugger>Breakpoints*）。

教程步骤 17：

要使用断点再次运行程序：

- 选择 *Debugger>Reset>Processor Reset* 复位应用程序。
- 双击以下代码行旁的灰色区域，设置一个断点：

```
incf COUNT,F ;increment count
```

- 选择 *Debugger>Run* 运行程序。程序执行应该在断点处停止。此时 Watch 窗口中所有项的值都应该为 0x00。可以从该断点处单步运行代码以观察 Watch 窗口中值的变化。

3.19 选择编程器

一旦对代码进行了调试并顺利运行之后，就应该为器件编程并尝试在应用中运行该程序。

通过 *Programmer>Select Programmer* 选择编程器。要使用的编程器取决于要编程的器件。

- **PICSTART Plus** 和 **PICkit 1** 开发编程器——用于在开发 （非生产）环境中为大多数 PICmicro MCU 器件编程。参见 MPLAB IDE 目录中的 Readme 文件以查看所支持的器件的列表。
- **MPLAB ICD 2** 在线调试器——用于为某些闪存器件编程。参见 MPLAB IDE 目录中的 Readme 文件以查看受支持的器件的列表。
- **MPLAB PM3** 器件编程器——用于为 PIC18 MCU 和 dsPIC DSC 器件编程。参见 MPLAB IDE 目录中的 Readme 文件以查看所支持的器件的列表。
- **PRO MATE II** 器件编程器——用于为大多数 PICmicro MCU 器件编程。参见 MPLAB IDE 目录中的 Readme 文件以查看所支持的器件的列表。

一旦选择了编程器，就会发现 IDE 发生了以下改变：

- Programmer 菜单——几个标准的选项将变为可用。还可能会出现其他特定于工具的选项。该菜单的大部分选项均会出现在源代码 （文件）窗口或 Program Memory 窗口的右击弹出菜单中。
- 编程器工具栏——几个标准的选项将变为可用。还可能会出现其他特定于工具的选项。
- 状态栏——现在应该在此显示出选定的编程器。

3.20 对器件进行编程

一般来说，要对器件编程：

- 在代码中或是通过 *Configure> Configuration Bits* 检查配置位的值。
- 在 *Programmer>Settings* 对话框中检查编程设置。
- 选择 *Programmer>Program*。

完成对器件的编程后，就可以尝试在应用中使用它了。若该器件不能按照预期运行，将需要调试代码并再次对该器件编程 （或对一个新器件编程），然后再次将器件插到应用中。

3.21 使用 MICROCHIP 帮助

Microchip Technology 提供有关 MPLAB IDE 及其所有开发工具的在线 HTML 帮助。选择 *Help>Topics* 在 Help Topics （帮助主题）对话框中会列出所有可用的帮助文件。

- 帮助查看导航
- 可用的帮助类型

3.21.1 帮助查看导航

帮助查看器的主要功能如图 3-2 所示。

图 3-2: 帮助查看器——主要功能

3.21.1.1 按钮栏

- 显示 / 隐藏——显示或隐藏左边的窗格。
- 返回——返回一个主题。
- 打印——打印当前显示的主题。
- 选项——显示更多可用的选项。

3.21.1.2 浏览按钮

每个在视图窗格中显示的主题都带有浏览按钮。

- 单击 **<**，根据目录中的排序，浏览前一个帮助主题。
- 单击 **>**，根据目录中的排序，浏览下一个帮助主题。

3.21.1.3 导航窗格

- 要查看将所有的 Microchip 帮助主题按功能组织的列表，请单击**目录**选项卡。然后单击文件夹旁边的 “＋” 号，打开该文件夹，以查看其中所包含的主题。单击列出的主题，即可在查看器右边的窗格中查看与该主题相关的帮助信息。
- 要查看帮助主题的按字母排序的列表，请单击**索引**选项卡。
- 要搜索帮助主题中的关键字，请单击**搜索**选项卡。
- 要创建和维护收藏主题的列表，请单击**书签**选项卡。
- 要关闭左边的窗格，请单击按钮栏上的**隐藏**按钮 （单击后它将变为**显示**按钮）。要查看左边的窗格，请单击按钮栏上的**显示**按钮 （单击后它将变为**隐藏**按钮）。

3.21.1.4 视图窗格

单击蓝色的带有下划线的文字以激活一个超级链接：

- 跳转到另一个主题。请单击**返回**按钮返回，或者浏览到其他主题。
- 打开一个弹出框。若要关闭该弹出框，请单击弹出框以外的区域。

3.21.2 可用的帮助类型

通过 MPLAB IDE Help 菜单即可使用工具的帮助文件。通过这些帮助项可获取有关特定对话框或窗口的帮助信息。工具的限制则可从 Settings 对话框中获取。

3.21.2.1 HELP 菜单

通过 Help 菜单即可使用每个开发工具的帮助文件。大多数帮助文件均包含以下内容：

- 工具概述
- 教程 / 使用示例
- 通用信息
- 疑难解答信息
- 参考资料

3.21.2.2 对话框

单击每个对话框中的 **Help** 按钮，即可使用对话框帮助。单击 Help 将显示使用对话框完成某项任务的逐步说明。

3.21.2.3 窗口

按下 **F1** 键，即可使用窗口帮助。当窗口处于激活状态时，按下键盘上的 **F1** 键，即可查看有关该窗口内容的信息。

3.21.2.4 限制

当选定了器件 （*Configure>Select Device*）并设置了调试工具 （*Debugger>Select Tool*）以后，就会显示调试工具限制。可以打开 Settings 对话框（*Debugger>Settings*）并单击 **Limitations** 选项卡查看这些设置。

可通过打开工具的帮助文件 （*Help>Topics*），并在 “目录”或 “索引”中查找 “Limitation”以获取调试和编程工具的限制。



第 4 章 项目和工作区

4.1 简介

项目和工作区是 MPLAB IDE v6.xx 的两个主要部分。

项目包含了编译应用程序所需的文件 （源代码和链接描述文件等），还包含这些文件与各种编译工具及编译选项之间的对应关系。

工作区包含了有关选定器件、调试工具和 / 或编程器、打开的窗口和窗口的位置及其他 IDE 配置设置的信息。

设置项目及其对应工作区的最好方法是使用项目向导。这样就可以在一个工作区中设置一个项目。

若要设置较高级的应用，可以手动设置项目和工作区。可以利用工作区设置，在一个工作区中打开多个项目。而且，还可以在多项目工作区中定制每个项目，以创建较大应用程序 （多项目并存）的一部分。

如果使用的是只有一个汇编文件的应用程序，则可以使用 Quickbuild （快速编译）（*Project>Quickbuild*）来通过 MPASM 汇编器汇编代码，而无须创建项目。但是仍需要有一个对应的工作区来保存设置。

项目

- 使用项目向导
- 创建 / 更新任何项目
- 设置项目结构——相关路径
- 项目文件夹和文件
- 使用版本控制系统 （VCS）
- 设置 / 更改项目

项目和工作区

- 使用单个项目和工作区 （如 MPLAB IDE v5.xx 和更早的版本）
- 在单个工作区中使用多个项目
- 编译无需项目的应用程序 （快速编译）

4.2 使用项目向导

项目向导由若干对话框组成，可以遵循这些对话框设置项目。

- 项目向导——欢迎
- 项目向导——选择器件
- 项目向导—— 选择语言工具包 （选择语言工具包）
- 项目向导——为项目命名
- 项目向导—— 添加文件
- 项目向导——摘要

4.2.1 项目向导——欢迎

选择 *Project>Project Wizard* 启动项目的设置向导。

请遵循 Project Wizard 中的对话框，设置新的项目。

还可以使用帮助查看器中的浏览顺序来查看向导中的步骤。

单击 **Next** 继续。

4.2.2 项目向导——选择器件

Step 1: Select a Device （**步骤 1**：选择器件）——将显示出默认的 MPLAB IDE 器件，或之前使用 Select Device 对话框选定的器件。从列表中为项目选择器件，或输入器件的名称。

单击 **Next** 继续。

4.2.3 项目向导—— 选择语言工具包

Step 2: Select a Language Toolsuite （**步骤 2**：选择语言工具包）

- 在 "Active Toolsuite"下拉菜单中为项目选择语言工具包。虽然无论选择什么器件都会显示第三方语言工具，但是在这个菜单中只能选择已安装且适用于所选器件的 Microchip 语言工具包。要查看所有可用的 （已安装的）工具包，请选择 "Show all installed toolsuites"（显示所有已安装的工具包）。

如果在这个列表中仍然找不到所需要的工具包，请单击 "**Help!My Toolsuite Isn't Listed!** "（需要帮助！没有列出我所需的工具包）查阅更多信息。

- "Toolsuite Contents"下的框中将显示所选工具包中的工具列表。前面有红 "X"的工具表示该工具未安装或者 MPLAB IDE 无法获得该工具的可执行文件的路径。要指定或检查工具所对应的可执行文件的位置分配，请单击工具，以在 "Location of Selected Tool"（所选工具位置）下显示出可执行文件及其路径。在该文本框中输入或修改路径，或者单击 **Browse** 查找可执行文件。

单击 **Next** 继续。

4.2.4 项目向导——为项目命名

Step 3: Name Your Project （**步骤 3**：为项目命名）—— 输入新项目的名称和位置。要设置目录：

- 输入现有目录的路径，或是输入新目录的路径。输入新的路径时，单击 **Next>** 会提示创建目录。
- 单击 **Browse** 找到现有的目录或是想要放置新目录的上一级目录。在 Browse for Folder 对话框中单击 **OK**。如果正在创建新目录，请在完成路径后单击 **Next>**。如果目录不存在将会提示创建目录。

4.2.5 项目向导—— 添加文件

Step 4: Add Any Existing Files to Your Project （**步骤 4**: 向项目中添加任何现有文件）—— 如果已经准备了想向新项目中添加的文件，现在将选择它们：

- 选择要添加的文件。单击一个文件名以选择该文件。按住 Ctrl 键并单击以选择多个文件。按 **ADD>>** 列出被添加到新项目的文件。
- 要将文件从列表中删除，请单击文件名选择要删除的文件，然后单击 **Remove**。

如果希望将文件复制到新项目而不是移动文件，请选中已添加的文件旁边的复选框。

单击 **Next** 继续。

4.2.6 项目向导——摘要

检查项目的摘要信息。如果有任何错误信息，请使用 **Back** 返回到需要更改信息的对话框。

当对项目设置满意时，单击 **Finish**。

4.3 创建 / 更新任何项目

无论使用的是单项目工作区还是多项目工作区，创建或更新项目的过程均相同。

1. 在当前工作区中创建 / 打开一个项目。项目就会出现在项目窗口中。
   - 选择 *Project>New*，创建新项目。在对话框中输入新项目的名称和位置。
   - 选择 *Project>Open*，打开现有项目。
2. 选择 *Project>Build Options>Project* 的 General 选项卡，指定 / 更改项目文件的路径。
3. 首先通过选择 *Project>Set Language Tool Locations* 指定每个语言工具的路径来指定 / 更改项目中的语言工具。然后选择 *Project>Set Language Toolsuite* 设置项目的工具包。
4. 通过选择 *Project>Build Options>Project*，或右击项目窗口中的项目名称并选择 Build Options，然后单击相应的语言工具选项卡，设置 / 更改项目的语言工具属性。

5. 选择 *Project>Add Files* 为项目添加文件，方法是右击项目窗口中的项目名称并选择 Add Files to Project （为项目添加文件），或右击项目窗口中的文件类型并选择 Add Files，添加特定类型的文件。

注：	即使文件尚不存在，也可以将其添加到项目中。这对于在真正开发这些文件之前构建项目是非常有用的。请切记在编译项目之前创建这些文件，否则编译肯定会失败。

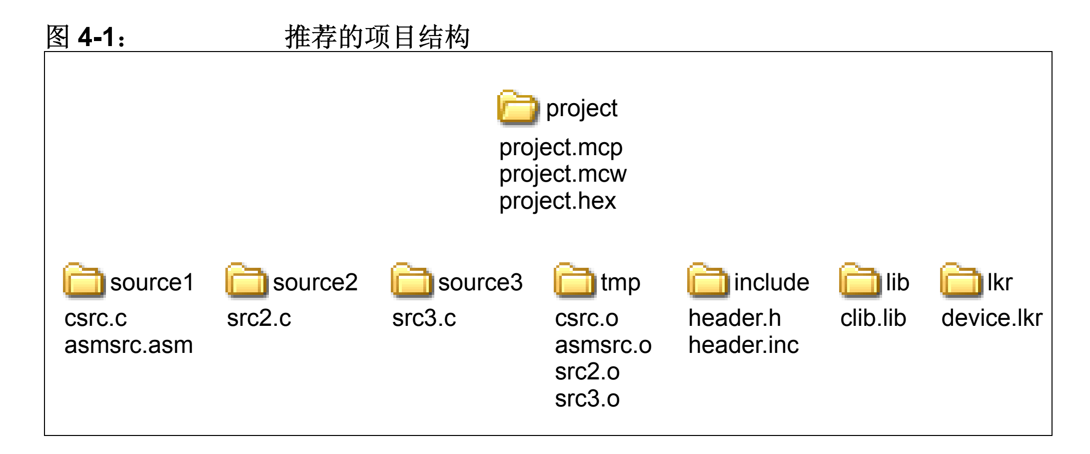
6. 选择 *Project>Remove File From Project* 或右击项目窗口中的文件并选择 Remove，删除项目中的文件。
7. 如果要使对单个文件所设置的语言工具属性不同于整个项目，请选择 *Project>Build Options> 文件名*，其中文件名是一个文件的名称，或者在项目窗口中右击文件名并选择 Build Options。这些设置仅应用于单个文件。

4.4 设置项目结构——相关路径

下面是创建推荐的项目目录结构的示例，该目录结构是使用 Build Options 的 General 选项卡上的相关路径创建的。保持所有项目子文件夹平行可以更容易地对中间文件、include 文件、库文件和链接描述文件分组。

注：	确保源代码文件使用惟一的名称，否则目标文件会被覆盖。例如，`file.asm`汇编后生成`file.o`，而`file.c`编译后也生成`file.o`，这样就会覆盖第一个 `file.o`。

图 4-1： 推荐的项目结构



Project>Build Options>Project， General 选项卡

输出目录：`C:\project`

中间文件目录：`..\tmp`

汇编器 Include 文件路径：`..\include`

Include 文件路径：`..\include`

库文件路径：`..\lib`

链接描述文件路径：`..\lkr`

4.5 项目文件夹和文件

打开项目时，会将其所有文件分配给以下某一文件夹中。与文件夹关联的过滤器决定文件夹中将包含哪些文件和文件夹的顺序。如果两个文件夹可接受同一个文件，则会将文件分配给第一个文件夹。

- 源文件——只有这些文件才能被工具包接受作为其文件命令的输入。示例文件：`file.asm`、`file.c`。
- 头文件——MPLAB IDE 在编译时不使用此类文件。可以在头文件中记录项目的相关信息，还可以很方便地访问这些文件。双击项目窗口中的文件名，就可以在编辑器中打开相应文件。示例文件：`header.inc`、`header.h`。
- 目标文件——此类文件是预编译的目标文件，而不是汇编或编译项目源文件后生成的目标文件。从功能上看，此类文件与库文件的区别不大，因为这两种文件都只在编译的最后阶段进行链接。示例文件：`ofile.o`。
- 库文件——工具包需要使用这个文件夹中的所有文件和目标文件，最后的链接步骤将包含这些文件。示例文件：`fft.o`。
- 链接描述文件——该文件夹中应该只有一个文件。如果用户有多个链接描述文件，那么只有第一个脚本有效。工具在链接步骤中将使用链接描述文件。示例文件：`linker.lkr`、`linker.gld`。
- 中间文件 （不会显示在项目窗口中）——项目保留这个文件夹，但不向用户显示该文件夹。在汇编和编译源文件时，将会在这个文件夹中创建将要用于链接的目标文件。链接步骤中将包含这个文件夹中的每个目标文件，以及目标文件夹和库文件文件夹中的文件。

注：	当用户执行 Clean 命令时，会删除这些文件。

- 输出文件 （不会显示在项目窗口中）——项目保留这个文件夹，但不显示。这些文件是链接步骤生成的最终输出文件。Clean 命令会删除所有这些文件。
- 其他文件——不适合归为上述类别中的任何文件都属于此类别。示例文件：`file.txt`, `stim.sts`。

4.6 使用版本控制系统 （VCS）

如果希望在开发应用代码时使用版本控制系统，现在就可以使用 Projects 菜单在 MPLAB IDE 中实现这一点。

- 选择 *Project>Version-Control*，设置版本控制应用程序，使之能在 MPLAB IDE 中使用 （参见第 **9.25** 节 **“Version-Control 对话框”**）。

注：	VCS 中必须已有文件夹 / 文件，以供 MPLAB IDE 在项目中使用这些文件。

要在 VCS 中保存整个项目 （和工作区）：

- 在 MPLAB IDE 外部 Check in 整个项目。当下一次在 MPLAB IDE 中 Check out 并打开项目时，就可在 MPLAB IDE 中看到 VCS 特定的信息了 （参见下文）。

要在 VCS 中仅保存某些项目文件：

- 如果希望在特定项目中使用 VCS 中包含的通用文件，则需要在 MPLAB IDE 外部 Check out 这些文件。
- 将 Check out 后的文件添加到项目。现在应可在 MPLAB IDE 中看到 VCS 特定的信息 （参见下文）。

一旦将项目设置为使用 VCS，就可以看到 MPLAB IDE 发生了下列变化：

- 项目窗口会显示项目文件的版本控制信息，即显示 Check out 状态。
- 在项目窗口中右击文件名，即可选择 VCS 文件的版本控制命令。
- 可以在 Project-Display Preferences 对话框中设置刷新选项，以便了解文件的 Check in/Check out 状态。
- 输出窗口会包含一个 Version Control 选项卡，显示 MPLAB IDE 和版本控制系统之间的动作。

下面列出了有关使用目前支持的版本控制系统的更多信息。

4.6.1 Microsoft Visual Source Safe

了解 VSS 和 MPLAB IDE 是在 MPLAB IDE 中使用 VSS 的必要条件。

欲知有关这个版本控制系统的更多信息，请参见相关网站。

- Microsoft Visual Source Safe：http://www.microsoft.com

要在 MPLAB IDE 中使用 VSS，需要有一些本地 （在 PC 机上） VSS 文件。如果要在网络驱动器上使用 VSS，则需要运行
NETSETUP.EXE，以在 PC 机上安装适当的文件。

4.6.2 PVCS

了解 PVSS 和 MPLAB IDE 是在 MPLAB IDE 中使用 PVSS 的必要条件。

欲知有关这个版本控制系统的更多信息，请参见相关网站。

- PVCS ：http://www.merant.com/Products/ECM/VM/home.asp

4.6.3 CVS

了解 CVS 和 MPLAB IDE 是在 MPLAB IDE 中使用 CVS 的必要条件。

欲知有关这个版本控制系统的更多信息，请参见相关网站。

- CVS：https://www.cvshome.org

4.6.3.1 导入文件

如果资源库中还没有源文件，则必须首先导入源文件。该操作将在资源库中创建目录结构并替换那里的所有文件。

警告
不要在资源库中放置工作区 （MCW）或编译状态 （MCS）文件。MPLAB IDE 在运行时需要写入这些文件。

下面给出了导入命令的示例。假定已经为 CVSROOT 环境变量指定了合适的值，因此未给出 -d 参数。需要在源树最顶层的目录中执行此命令。此命令将自动递归到子目录。

```
$ cd TopOfSrcTree
$ cvs import -m “message” DirWithinCVSRoot VendorTag ReleaseTag
```

4.6.3.2 Checkout 文件

即使在本地具有源文件，仍需要执行一次 cvs checkout （首次使用时），才能在 MPLAB IDE 中用 CVS Check out 文件。这一 Check out 操作会创建 CVS 在各种文件上执行命令所需的管理文件。所有这些管理文件均存储在源树中每个目录下的隐藏子目录中。每个隐藏子目录的名称均为 CVS。

试图在不具备这些必需的 CVS 目录的情况下，对现有源代码文件执行 Check out 操作，就会导致 CVS 报错。在执行 Check out 之前，应删除源代码文件的本地副本，或将它们移动到其他位置。

下面列出了执行 Check out 操作的示例：

```
$ cd DirABOVEWhereIWantMySrcTree
$ cvs checkout MyProject
cvs server: Updating MyProject
U MyProject/main.c
U MyProject/such-and-such-a-file.c
U MyProject/foo.h
```

由于 CVS 实际上会在当前工作目录中创建整个源树，因此必须在希望放置源树的目录的*上层目录*中执行该命令，这一点很重要。

4.6.3.3 添加新目录

只要资源库中具备了必需的目录结构，就可在 MPLAB IDE 中使用 cvs add 命令来添加新的源代码文件。如果资源库中缺少任何必需的目录， add 命令均不会添加任何文件。

要在资源库中创建目录结构，必须再次使用 cvs import 命令。下面列出了该命令的示例：

```
$ cd TopOfSrcTree
$ cvs import -m "message" MyProject/NewSubDir VendorTag ReleaseTag
N MyProject/NewSubDir/yet-another-file.h

No conflicts created by this import（导入操作未产生任何冲突）
```

同样，必须在源树的最上层目录中执行这个命令。不过这一次，当指定资源库中的目录时，应给出要在资源库中添加的新目录的路径。（在本例中该路径由 MyProject/NewSubDir 表示。）

4.7 设置 / 更改项目

可通过以下步骤来手动创建新项目 （推荐使用 Project Setup Wizard （项目设置向导）），或更改现有的项目设置。

- 创建新项目
- 新建项目窗口
- 设置项目——设置语言工具
- 设置项目——选择工具包
- 设置项目——添加文件
- 设置项目——设置编译选项
- 设置项目——保存并编译

4.7.1 创建新项目

要在当前工作区中创建新项目：

- 使用 “Windows 资源管理器”为项目创建一个目录 （例如 `C:\Proj1`）。
- 在 MPLAB IDE 中，选择 *Project>New*。New Project （新建项目）对话框会打开，提示您输入新项目的名称和位置。
- 输入项目名称，例如 `Proj1`。
- 输入新项目的路径。单击 Browse 找到目录。
- 单击 **OK**。将打开 Save Workspace As （将工作区另存为）对话框。

要保存包含新项目的工作区：

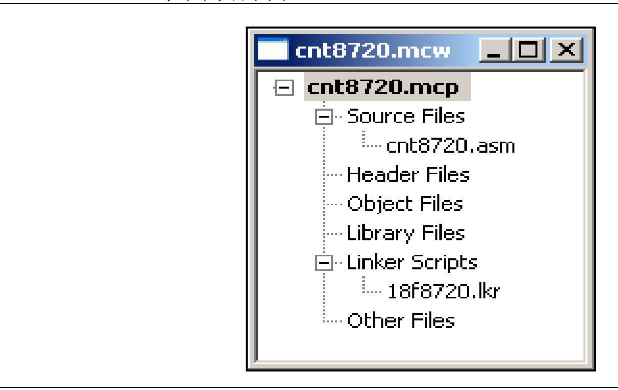
- 输入工作区的名称，即 `cnt452`。
- 单击 **OK**。

MPLAB IDE 将创建扩展名为 `.mcp` 的项目文件和扩展名为 `.mcw` 的工作区文件。

4.7.2 新建项目窗口

如果项目窗口还未打开，可通过选择 *View>Project* 将其打开。工作区的名称应显示在项目窗口最顶上的栏中。项目名称将显示在项目窗口内的最顶端。在其下方会显示一个列出文件类型的树结构。

图 4-2： 示例项目窗口



4.7.3 设置项目——设置语言工具

要添加项目中使用的语言工具，请选择 *Project>Set Language Tool Locations*，或在项目窗口中右击项目名称并从弹出菜单中选择 Set Language Tool Locations （设置语言工具的位置）。

- 单击工具包中的已注册工具，选择它。
- 输入 “位置”，或单击 **Browse** 找到工具的可执行文件的位置。
- 对希望在项目中使用的每个工具重复此操作。
- 完成后，单击 **OK**。

现在需要为项目指定工具包。

4.7.4 设置项目——选择工具包

要选择项目中使用的工具包，请选择 *Project>Set Language Toolsuite*，或在 Project 项目窗口中右击项目名称并从下拉菜单中选择 Set Language Tool-suite （设置语言工具包）。

- 从 “Active Toolsuite”列表中选择项目工具包。
- 单击 **OK**。

使用的工具包取决于将要为其开发代码的器件。

- Microchip 语言工具——用于 PICmicro MCU、 Microchip 存储器和 KEELOQ 器件
- dsPIC 语言工具——用于 dsPIC 器件
- 第三方工具——用于各种 Microchip 器件

4.7.5 设置项目——添加文件

现在将为项目添加文件。项目中包含的基本文件类型可以通过单击项目窗口中项目名称左边的 “+”号查看。

> **注：** 可以输入尚不存在的文件名。但是，在编译项目之前，必须使用诸如 MPLAB 编辑器的文本编辑器创建这个文件。

要在项目中插入文件，请选择 *Project>Add Files*，或右击项目窗口中的项目名称并从下拉菜单中选择 “Add Files”。此外，若已知要添加的文件的类型，可以在项目窗口中右击这个类型，并从下拉菜单中选择 “Add Files”（添加文件）。

- 将打开 Add Files to Project 对话框。
- 单击一个文件，或输入文件名，选择要插入的文件。如果所需的文件没有列出，请打开 Files of Type （文件类型）下面的下拉菜单，选择合适的文件类型。如果不确定文件类型，请选择所有文件 （*.*）。
- 要选择将要插入的多个文件，请在按住 Shift 键的同时单击鼠标选择连续的文件，或在按住 Control 键的同时单击鼠标选择不连续的文件。
- 单击 **Open**。

4.7.6 设置项目——设置编译选项

MPLAB IDE 对项目中包含的工具和文件具有默认设置。然而，您可能想要或需要修改这些设置。

要设置编译选项，请选择 *Project>Build Options>Project* 或在项目窗口中右击项目，并在弹出菜单中选择 Build Options。

- 将打开 Build Options）对话框。
- 单击 **General** 选项卡，输入文件的路径或者单击 **Browse** 按钮查找路径，这些文件可能是输出文件、 include 文件、库文件或链接描述文件。

> **注：** 这些是 MPLAB IDE 路径。并不是所有的语言工具都会使用此信息。

- 单击具体的语言工具的选项卡 （如 MPASM Assembler）并设置该工具的操作特性。
- 单击 **OK**。

要改写某个特定项目文件的项目设置，例如 `ProjFile1.asm`，请选择 `Project>Build Options>ProjFile1.asm`，或右击项目窗口中的 `ProjFile1.asm`，并从弹出菜单中选择 Build Options。

- 将打开 File Settings （文件设置）对话框。
- 单击具体的语言工具的选项卡 （如 MPASM Assembler）并设置该工具的操作特性。
- 单击 **OK**。

4.7.7 设置项目——保存并编译

此时，应该选择 *Project>Save* 或右击 项目窗口中的项目名称并选择 “Save Project”（保存项目）来保存项目。

现在已经准备就绪，可以编译项目了。右击项目窗口中的项目名称，并从弹出菜单中选择 “Build All”。

- 如果选用了 MPASM 汇编器，就会打开一个状态窗口，显示编译的进度和最终的结果。编译完成后将关闭该窗口。
- 还将打开 Output 窗口。该窗口将显示有关编译的信息，包括遇到的任何错误。

如果项目编译失败，请检查以下各项，然后重新编译项目：

- 检查在编辑器窗口中输入的所有代码的拼写和格式。如果在 Output 窗口中报告有错，双击该错误就会在源代码窗口左侧的灰色区域中以绿色箭头指出源代码中出错的行。
- 检查用于该项目及其项目文件的语言工具是否正确。

编译成功后，将装载由语言工具生成的调试文件 （`*.cod`、 `*.cof` 或 `*.elf` 等）。该文件允许您调试源代码，并在 Watch 窗口中直观地查看变量。

4.8 使用单个项目和工作区

MPLAB IDE 中最常见的应用程序开发配置就是在单个工作区中使用单个项目。如果希望使用 MPLAB IDE v5.xx 样式的项目 （不考虑工作区），请选择 *Configure>Settings* 的 **Projects** 选项卡，并选中 “Use one-to-one project-workspace model”（使用一对一的项目—工作区模型）。

1. 选择用于开发的器件 （*Configure>Select Device*）。
2. 在当前工作区中创建 / 打开项目。
3. 编译项目。
   - 要编译整个项目，请选择 *Project>Build All* 或右击项目窗口中的 项目名称并选择 Build All。
   - 要仅编译项目中经过更改的文件，请选择 *Project>Make*，或右击项目窗口中的项目名称并选择 Make。

现在已准备就绪，可以调试应用程序了。当调试会话完成后：

- 选择 *Project>Save Project*，然后选择 *Project>Close*，关闭项目及其关联的工作区。
- 选择 *Project>Open*，重新打开项目及其关联的工作区。

4.9 在单个工作区中使用多个项目

您可能希望在单个工作区中使用多个项目。这使多个项目使用相同的工作区设置 （即选定器件、调试工具和窗口等）。

在单个工作区中使用多个项目对某些应用来说比较有利。参见以下示例。

例 4-1： 引导装载程序

引导装载程序具有自己的源代码，它的唯一功能是将应用程序的新版本下载并编程到闪存存储器中。引导装载程序总是位于程序存储器中的固定位置，可以通过应用程序的特定功能执行。引导装载程序可以通过通信端口下载程序，并用新版本覆盖现有的应用程序。由于引导装载程序和应用程序位于同一器件，因此它们共享 dsPIC30F 器件的相同设置，但是它们是互相独立的代码段。应用程序仅知道执行引导装载程序的入口地址。引导装载程序仅知道升级代码的起始地址。

例 4-2： 测试

多个项目的另一种用途是开发单个应用程序的一系列测试。这些测试可能位于程序存储器中的相同位置，但决不会同时加载。

例 4-3： 应用程序配置

第三种可能性是应用程序的最终产品具有多种配置。应用程序可能会以多种商标或多种语言售出。应用程序的商标信息或语言信息包含在每个项目的独立资源文件中。

4.9.1 设置多个项目

要设置工作区使用多个项目：

1. 选择用于开发的器件 （*Configure>Select Device*）。
2. 在当前工作区中创建 / 打开第一个项目。
3. 确保选择了下列选项。然后对希望添加到工作区的每个项目重复步骤 2。
   - *Configure>Settings*， **Projects** 选项卡：去掉对 “Use one-to-one project-workspace model”（使用一对一的项目—工作区模型）的选择。
   - *Configure>Settings*， **Program Loading** 选项卡：选中 “Clear program memory upon loading a program”（加载程序时清空程序存储器）
4. 选择 *Project>Set Active Project>projectname.mcp*，其中 `projectname.mcp` 是项目名称，或右击项目窗口中的项目并选择 Set As Active Project （设置为活动项目），激活项目。项目名称将在项目窗口中以粗体显示。
5. 编译活动项目
   - 要编译整个项目，请右击项目窗口中的项目名称，并选择 Build All。
   - 要仅编译项目中经过更改的文件，请右击项目窗口中的项目名称并选择 Make。

现在已准备就绪，可以调试应用程序了。当调试会话完成后：

- 选择 *Project>Save Project*，或右击项目窗口中的项目并选择 Save Project，保存当前活动的项目。
- 选择另一个项目作为活动项目，编译此项目并进行调试。
- 选择 *Project>Close*，或右击项目窗口中的项目并选择 Close Project，从工作区中删除该活动项目。
- 选择 *File>Close Workspace*，关闭并保存所有项目及其关联的工作区。该操作会提示您进行保存。
- 选择 *File>Open Workspace*，重新打开所有项目及其关联的工作区。

4.9.2 确定活动项目

活动项目是工作区中下列 Project 菜单和工具栏命令可以起作用的项目：

- Clean
- Find in Project Files
- Save Project
- Add Files to Project
- Select Language Toolsuite

所有其他 Project （项目）菜单、工具栏和鼠标右键命令会作用于选择命令时已指定的项目。举例来说，如果在项目窗口中右击了一个非活动的项目，可以通过选择 Build All 或 Make 来编译此项目，此项目的编译结果将会被加载到程序存储器中。

为了避免混乱，建议将需要开发和编译的项目作为活动项目。

工作区中最后一个打开的项目会自动设置为活动项目。要将项目设置为活动项目，请选择 *Project>Set Active Project> project-name.mcp*，其中 projectname.mcp 是项目的名称，或右击 项目窗口中的项目并选择 Set As Active Project。项目名称将在 Project 项目窗口中以粗体显示。

4.9.3 使用并存项目

可能希望具备多个项目，并使用这些项目创建代码以占用程序存储器的不同区域，从而创建一个应用程序。以这种方式使用的项目被称为并存项目。

要创建并存项目的工作区：

1. 开始使用多项目工作区。
2. 清除对 *Configure>Settings* 的 **Program Loading** 选项卡中的 “Clear program memory upon loading a program”选项的选择。

4.10 编译无需项目的应用程序

如果仅想使用 MPASM 汇编器汇编单个汇编文件，则不需要使用项目，但要使用 Quickbuild。

1. 选择用于开发的器件 （*Configure>Select Device*）。
2. 确保 Quickbuild 可用 （*Project>Set Active Project>None* （快速编译模式））。
3. 打开一个文件 （编辑器）窗口并输入汇编代码 （*File>New*），打开现有汇编文件 （*File>Open*），或单击打开的代码编辑窗口，激活窗口。

注：	如果文件窗口处于非活动状态，菜单上的 Quickbuild 选项将灰掉。

4. 选择 *Project>Quickbuild(file.asm)* 汇编应用程序，其中 file.asm 是活动的汇编文件的名称。

现在已准备就绪，可以调试应用程序了。当调试会话完成后：

- 选择 *File>Save Workspace*，将当前的 MPLAB IDE 配置保存到一个文件中。
- 选择 *File>Open Workspace*，恢复 MPLAB IDE 配置，以用于另一个调试会话。

注：



第 5 章 集成工具

5.1 简介

MPLAB IDE 设计为与许多 Microchip 和第三方开发工具配合工作。

语言工具

- 语言工具包
- Microchip 语言工具
- 第三方语言工具

软件 / 硬件工具

- 编辑器
- 软件模拟器
- 在线仿真器
- 在线调试器
- 编程器
- 第三方工具

5.2 语言工具包

MPLAB IDE 支持许多语言工具包。集成在 MPLAB IDE 中的是 Microchip MPASM 工具包，但是也可以使用许多其他的工具包，包括 Microchip C17、C18 和 C30 工具包，以及来自 HI-TECH、 IAR、 CCS、 microEngineering Labs 和 Byte Craft 的语言工具。有两种方法可以将这些工具集成到 MPLAB IDE 中：使用由制造商设计的插件，或者使用传统的可以为任何语言工具包定制的 “.MTC”文件。

语言工具包插件

语言工具的供应商可以编写一个可安装到 MPLAB IDE 中的插件，从而提供定制的对话框来访问各种工具编译选项。这是提供支持的首选方法。

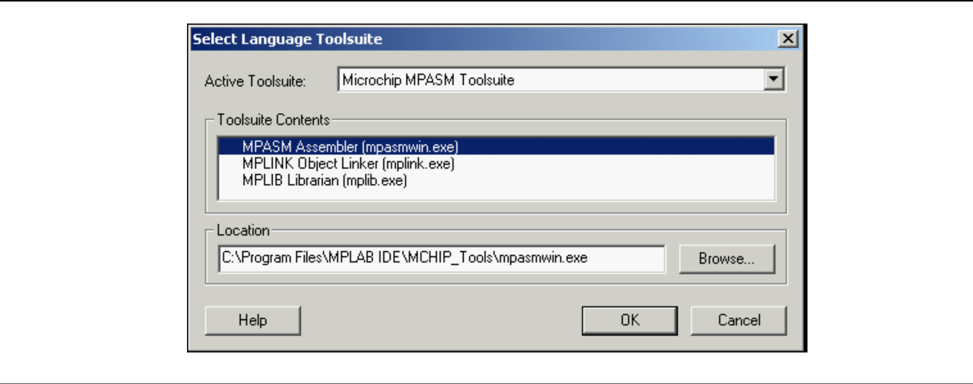
.MTC 文件

使用具有 `.mtc` 扩展名的特殊文件是集成语言工具的常规方法。使用这种方法，这些文件会被集成到一个对话框中以提供某种复选框，通过这种复选框即可使用语言工具的各种编译选项。使用这些文件可以控制语言工具包的大多数功能，但并非所有功能。

5.2.1 选择语言工具包

打开工作区之后，可以从 *Project>Select Language Toolsuite* 选择工具包。在某些情况下， MPLAB IDE 知道工具包的各个可执行文件的位置，但是有时必须先手动设定这些文件的位置， MPLAB IDE 才能使用这些工具。

图 5-1： 选择一个工具包



如果在工具包组件左侧有红色的 X，则意味着 MPLAB IDE 不知道可执行文件的位置。您必须输入完整的路径和可执行文件的名称或者浏览到正确的可执行文件。

5.2.2 查找 Microchip 语言工具包

Microchip 工具包及其位置如下所述。

Microchip MPASM 工具包

此工具包包括语言工具 MPASM 汇编器、 MPLINK 目标链接器和 MPLIB 目标库管理器。这些工具的可执行文件分别是：`mpasmwin.exe`、 `mplink.exe` 和 `mplib.exe`，它们位于 MPLAB IDE 主安装目录下的 `mchip_tools` 子目录中。欲知有关这些工具的更多信息，请参见 Microchip PICmicro 语言工具。

Microchip C17 工具包 （过时产品）

此工具包包含来自 Microchip MPASM 工具包的所有语言工具以及 MPLAB C17 C 编译器。 MPLAB C17 的安装独立于 MPLAB IDE。 MPLAB C17 的可执行文件是 `mcc.exe`，默认情况下它安装在 `c:\mcc\bin` 目录中。欲知更多有关这些工具的信息，请参见 Microchip PICmicro 语言工具。

Microchip C18 工具包

此工具包包含来自 Microchip MPASM 工具包的所有语言工具以及 MPLAB C18 C 编译器。 MPLAB C18 的安装独立于 MPLAB IDE。 MPLAB C18 的可执行文件是 `mcc18.exe`，默认情况下它安装在 `c:\mcc18\bin` 目录中。欲知更多有关这些工具的信息，请参见 Microchip PICmicro 语言工具。

Microchip ASM30 工具包

此工具包包含语言工具 MPLAB ASM30 汇编器、 MPLAB LINK30 目标链接器和 MPLAB LIB30 目标库管理器。这些文件的可执行文件是：`pic30-as.exe`、 `pic30-ld.exe` 和 `pic30-ar.exe`，它们位于 MPLAB IDE 主安装目录下的 `dspic_tools\bin` 子目录中。欲知更多有关这些工具的信息，请参见 Microchip dsPIC 语言工具。

Microchip C30 工具包

此工具包包含来自 Microchip ASM30 工具包的所有语言工具以及 MPLAB C30 C 编译器。 MPLAB C30 的安装独立于 MPLAB IDE。 MPLAB C30 的可执行文件是 `pic30-gcc.exe`，默认情况下它安装在 `c:\pic30\bin` 目录中。欲知更多有关这些工具的信息，请参见 Microchip dsPIC 语言工具。

5.3 MICROCHIP 语言工具

Microchip Technology 目前提供的语言工具包括：

- Microchip PICmicro 语言工具——支持 PICmicro MCU、 KEELOQ 和存储器件。
- Microchip dsPIC 语言工具——支持 dsPIC DSC 器件。

支持 Microchip 器件的其他语言工具可从第三方获得。如需这些供应商的完整列表，请访问我们的网站。

5.3.1 Microchip PICmicro 语言工具

Microchip 工具包支持 PICmicro 单片机 （MCU）、 Microchip 存储器和 KEELOQ 器件。（如需获得所支持的器件的列表，请参阅 MPLAB IDE 安装目录中的相关 Readme 文件。）

表 5-1： MICROCHIP 工具包支持

工具	名称	可执行文件
C 编译器——PIC18X 器件	MPLAB C18 C 编译器	`mcc18`
C 编译器——PIC17C 器件 （过时）	MPLAB C17 C 编译器	`mcc`
汇编器	MPASM 汇编器 （MPASM 汇编器）	`mpasmwin`
链接器	MPLINK 目标链接器	`mplink`
库管理器	MPLIB 目标库管理器	`mplib`

还可使用其他的命令行工具可执行文件和实用程序。欲知更多信息，请参阅语言工具文档。

帮助文件

从 MPLAB IDE Help 菜单，选择 Topics，然后在 Language Tools 下选择以下一种语言工具：

- MPASM Assembler
- MPLINK Object Linker （包含 MPLIB 目标库管理器）

文档

请在 MPLAB IDE 光盘 （DS51123）或 Microchip 网站上查找下列 C 编译器文档。

- 《MPLAB C18 C 编译器入门》（DS51295E_CN）
- 《MPLAB C18 C 编译器用户指南》（DS51288C_CN）
- MPLAB C18 C Compiler Libraries （DS51297）
- MPLAB C17 C Compiler User's Guide （DS51290）
- MPLAB C17 C Compiler Libraries （DS51296）

5.3.2 Microchip dsPIC 语言工具

Microchip dsPIC 工具包支持 dsPIC 数字信号控制器 （DSC）。（如需获得所支持的器件的列表，请参阅 MPLAB IDE 安装目录中的相关 Readme 文件。）

表 5-2： MICROCHIP dsPIC 工具包

工具	名称	可执行文件
C 编译器	MPLAB C30	`pic30-gcc`
汇编器	MPLAB ASM30	`pic30-as`
链接器	MPLAB LINK30	`pic30-ld`
归档器 / 库管理器	MPLAB LIB30	`pic30-ar`

还可使用其他的命令行实用程序。欲知更多信息，请参阅语言工具文档。

帮助文件

从 MPLAB IDE Help 菜单，选择 Topics，然后在 Language Tools 下选择以下一种语言工具：

- MPLAB ASM30
- MPLAB LINK30 （包含 MPLAB LIB30 和其他实用程序）

文档

请在 MPLAB IDE 光盘 （DS51123）或 Microchip 网站上查找下列 dsPIC DSC 文档。

- 《dsPIC30F 系列参考手册》（DS70046C_CN）
- dsPIC30F Programmer's Reference Manual （DS70030）

请在 MPLAB IDE 光盘 （DS51123）或 Microchip 网站上查找下列 dsPIC 语言工具文档。

- 《dsPIC 语言工具入门》（DS70094C_CN）
- MPLAB ASM30, MPLAB LINK30 and Utilities User's Guide （DS51317）
- 《MPLAB C30 C 编译器用户指南》（DS51284C_CN）
- dsPIC30F Language Tools Libraries （DS51456）
- dsPIC30F Language Tools Quick Reference Card （DS51322）

5.4 第三方语言工具

当前在 MPLAB IDE 中支持的第三方语言工具有：

- B Knudsen Data 语言工具
- Byte Craft 语言工具
- CCS 语言工具
- HI-TECH 语言工具
- IAR 语言工具
- microEngineering Labs 语言工具

支持 Microchip 器件的其他语言工具可从第三方获得。如需这些供应商的完整列表，请访问 Microchip 网站。

5.4.1 B Knudsen Data 语言工具

MPLAB IDE 支持下列 B Knudsen Data 语言工具：

工具	名称	可执行文件
用于 PIC12/14/16 器件的 C 编译器	CC5X	`cc5x`
用于 PIC18 器件的 C 编译器	CC8E	`cc8e`

欲知更多信息，请访问 B Knudsen Data 网站 （http://www.bknd.com）。

5.4.2 Byte Craft 语言工具

MPLAB IDE 支持下列 Byte Craft 语言工具：

工具	名称	可执行文件
用于 PIC12/14/16/17 器件的汇编器 /C 编译器	MPC	`mpc`

欲知更多信息，请访问 Byte Craft 网站 （http://www.bytecraft.com）。

5.4.3 CCS 语言工具

MPLAB IDE 支持下列 Custom Computer Services （CCS）语言工具：

工具	名称	可执行文件
用于 PIC12/16 （12 位）器件的命令行 C 编译器	PCB	`pcb`
用于 PIC12/14/16 （14 位）器件的命令行 C 编译器	PCM	`pcm`
用于 PIC18 器件的命令行 C 编译器	PCH	`pch`
用于 PIC12/14/16 器件的 Windows C 编译器	PCW	`pcw`
用于 PIC12/14/16/18 器件的 Windows C 编译器 （带有可用于 dsPIC 器件的扩充程序）	PCWH	`pcwh`

欲知更多信息，请访问 CCS 网站 （http://www.ccsinfo.com）。

5.4.4 HI-TECH 语言工具

MPLAB IDE 支持下列 HI-TECH 语言工具：

工具	名称	可执行文件
C 编译器、汇编器和链接器——PIC12/14/16 器件	PICC	picc
C 编译器、汇编器和链接器——PIC18 器件	PICC18	picc18
C 编译器、汇编器和链接器——对存储器有限制	PICC Lite	picl
C 编译器、汇编器和链接器——对功能有限制	PICC Demo	picc
C 编译器、汇编器和链接器——dsPIC 器件	dsPICC	dspicc

要在项目中使用 HI-TECH 语言工具，需做如下设置：

- 选择 <u>*Project>Set Language Toolsuite*</u>。在 Select Language Toolsuite 对话框中，选择支持 PIC12/14/16 器件的 HI-TECH PICC 工具包或支持 PIC18 器件的 HI-TECH PICC18 工具包。单击 **OK** 或单击 **Set Tool Locations** 继续设置。

> 注：　使用 PICC Lite 或 Demo 的用户，请选择 HI-TECH PICC 工具包。

- 如果您关闭了先前的对话框，请选择 <u>*Project>Set Language Tool Locations*</u>。在 Set Language Tool Location （设置语言工具位置）对话框的列表中找到将使用的工具包，并单击将使用的工具包中的第一个工具。
- 在 Location of Selected Tool （选定工具的位置）文本框中查看可执行文件的路径。如果未列出位置 / 路径，请输入或浏览到一个位置 / 路径。

> 注：　PICC Lite 用户，请输入 / 浏览到与在使用的 HI-TECH 工具对应的可执行文件，即，使用 `picl.exe` 代替 `picc.exe`。

- 单击 **OK**，直到关闭了所有的语言工具对话框为止。

欲知更多信息，请访问 HI-TECH 网站 （http://www.htsoft.com）。

5.4.5 IAR 语言工具

MPLAB IDE 支持下列 IAR 语言工具：

工具	名称	可执行文件
用于 PIC16/17 器件的 C 编译器	IAR PICmicro 16/17 C 编译器	iccpic
用于 PIC16/17 器件的汇编器	IAR PICmicro 16/17 C 编译器	apic
用于 PIC18 器件的 C/EC++ 编译器	IAR PICmicro PIC18 C 编译器	iccpic18
用于 PIC18 器件的汇编器	IAR PICmicro PIC18 C 编译器	apic18
用于 dsPIC 器件的 C/EC++ 编译器	IAR dsPIC C 编译器	iccdspic
用于 dsPIC 器件的汇编器	IAR dsPIC C 编译器	adspic
链接器	IAR 链接器	xlink

欲知更多信息，请访问 IAR 网站 （http://www.iar.com）。

5.4.6 microEngineering Labs 语言工具

MPLAB IDE 支持下列 microEngineering Labs 语言工具：

工具	名称	可执行文件
Basic 编译器，适用于大多数 PICmicro 器件	PicBasic 编译器	pb
Basic 编译器，适用于所有的 PICmicro 器件	PicBasic Pro 编译器	pbp

欲知更多信息，请访问 microEngineering Labs 网站 （http://www.melabs.com）。

5.5 编辑器

MPLAB IDE 支持 MPLAB 编辑器。

帮助文件

从 MPLAB IDE Help 菜单，选择 Topics，然后在 System （系统）下选择：

- MPLAB Editor

相关菜单

- File （文件）
- Edit （编辑）

5.6 软件模拟器

支持下列 Microchip 软件模拟器：

软件模拟器名称	模拟的器件 *
MPLAB SIM 软件模拟器	PICmicro MCU 和 dsPIC DSC

* 如需获得所支持的器件的列表，请参阅 MPLAB IDE 安装目录中的相关 Readme 文件。

帮助文件

从 MPLAB IDE Help 菜单，选择 Topics，然后在 Debuggers 下选择：

- MPLAB SIM

相关主题

- 第三方工具

5.7 在线仿真器

支持下列 Microchip 在线仿真器：

仿真器名称	仿真的器件 *
MPLAB ICE 2000	PICmicro MCU
MPLAB ICE 4000	PIC18 MCU 和 dsPIC DSC

* 如需获得所支持的器件的列表，请参阅 MPLAB IDE 安装目录中的相关 Readme （文件。

帮助文件

从 MPLAB IDE Help 菜单，选择 Topics，然后在 Debuggers 下选择：

- MPLAB ICE 2000
- MPLAB ICE 4000

相关主题

- 第三方工具

5.8 在线调试器

支持下列 Microchip 在线调试器：

调试器名称	支持的器件 *
MPLAB ICD 2	PICmicro 闪存 MCU 和 dsPIC 闪存 DSC

* 如需获得所支持的器件的列表，请参阅 MPLAB IDE 安装目录中的相关 Readme 文件。

帮助文件

从 MPLAB IDE Help 菜单，选择 Topics，然后在 Debuggers 下选择：

- MPLAB ICD 2

5.9 编程器

支持下列 Microchip 编程器：

编程器名称	可编程的器件 *
MPLAB PM3 器件编程器	PICmicro MCU 和 dsPIC DSC
PRO MATE II 器件编程器	PICmicro MCU、存储器件和 KEELOQ 器件
PICSTART Plus 开发编程器	大多数 PICmicro MCU
PICkit 1 开发编程器	选定的 PICmicro MCU
MPLAB ICD 2 在线调试器	PICmicro 闪存 MCU 和 dsPIC 闪存 DSC

* 如需获得所支持的器件的列表，请参阅 MPLAB IDE 安装目录中的相关 Readme 文件。

帮助文件

从 MPLAB IDE Help 菜单，选择 Topics，然后在 Programmers 下选择以下一种编程器：

- MPLAB PM3
- PRO MATE II
- PICSTART Plus

此外，还可以在 Debuggers 下选择：

- MPLAB ICD 2

文档

请在 MPLAB IDE 光盘 （DS51123）或 Microchip 网站上查找以下文档。

- PICkit 1 Flash Starter Kit User's Guide （DS40051）

相关主题

- 第三方工具

5.10 第三方工具

支持 Microchip 器件的其他开发工具可从第三方获得。如需这些供应商的完整列表，请访问 Microchip 网站 （www.microchip.com）。



第 6 章 MPLAB IDE 疑难解答

6.1 简介

本章旨在帮助您解决使用 MPLAB IDE 过程中所遇到的任何问题、错误或难点。若本章中没有可帮助您的信息，请参见前言中的联系方式联系 Microchip 。

本章包含以下主题：

- 常见问题 /FAQ ——最常遇到的和最常问到的问题的列表。
- 错误消息 ——由工具产生的错误消息的列表。
- 限制 ——工具的限制。

6.2 常见问题 /FAQ

- 使用 MPLAB IDE 时我不能获取语言工具。
- 我无法看到显示窗口中的所有字符。
- 最近我将项目从一台计算机移动到了另一台计算机。但它们不能工作或者仅工作一会儿，现在又不能工作了。
- Project 菜单和工具栏中的一些项灰掉了。
- 在单步运行高级语言代码时，如果单步进入一个没有源代码的函数，就会开始跳出一大串 Program Memory 窗口。
- 我不能找到所有的项目文件。
- 我的程序不断复位，为什么？
- 我不能暂停单步连续运行。
- 我尝试 Step Over / Step Out 一个函数 / 子程序，但现在我的程序被挂起了。

使用 MPLAB IDE 时我不能获取语言工具。

MPLAB IDE v6.xx 与 MPLAB IDE v5.xx 以及更早的版本有许多不同。可能需要进行升级，以使用新版本的 MPLAB IDE。请访问我们的网站或第三方网站以获取最新版本的语言工具。

我无法看到显示窗口中的所有字符。

当使用 “系统显示字体大小” 中的 “大字体”（标称值为 125%）并选择斜体字体时，在窗口右上角的一些字符可能会被截断。源文件编辑器使用的 Tetradyne SourceView 编辑器控制支持下列字体：Courier、 Courier New 和 Fixedsys。

最近我将项目从一台计算机移动到了另一台计算机。但它们不能工作或者仅工作一会儿，现在又不能工作了。

项目文件 （.mcp）是可以移动的。工作区文件 （.mcw）可能可以移动也可能不可以移动。若想在操作系统不同或硬件配置不同的计算机间移动项目和工作区，将需要升级您的工作区或者创建一个新的工作区。

请查看 Saved Information （已保存信息）以了解更多保存在这些文件中的信息。

Project 菜单和工具栏中的一些项灰掉了。

Project 菜单和工具栏中的一些右键相关项需在设置一个活动项目后才可使用。请检查 *Project>Set Active Project*，确认没有选择 None (Quickbuild)。在项目窗口中，活动项目名显示为黑体。

在单步运行高级语言代码时，如果单步进入一个没有源代码的函数，就会开始跳出一大串 Program Memory 窗口。

如果单步进入一条指令，而该指令没有相关源代码行 （如库函数），因此在程序存储器窗口中没有相关项， MPLAB IDE 将会打开另一 Program Memory 窗口。

要避免该问题，请先选择 *Configure>Settings*, **Debugger** 选项卡。然后选中 “Browse for source if file not found”（如果未找到文件请浏览源代码）和 “Show disassembly if source is unavailable”（如果源代码不可用，显示反汇编）。

我不能找到所有的项目文件。

请确认已将项目目录结构设置为平行方式，或把目标文件归类到与源文件相同的目录下，而不是项目的 tmp 目录下。

还需确保源文件使用了唯一的文件名 （`file1.asm` 和 `file2.c`, 而不是 `file.asm` 和 `file.c`），这样就不会生成相同文件名的目标文件，具有相同文件名的目标文件将会相互覆盖。

我的程序不断复位，为什么？

请检查选定器件的配置位设置 （*Configure>Configuration Bits*）。一些复位功能，如 Watchdog Timer Reset （看门狗定时器复位）默认情况下是使能的。

我不能暂停单步连续运行。

要暂停单步连续运行，请使用菜单选项 *Debugger>Halt* 而不是工具栏上的 Halt 或快捷键 **F5**。

我尝试 Step Over / Step Out 一个函数 / 子程序，但现在我的程序被挂起了。

单步跳过或跳出函数或子程序实际上会执行该函数 / 子程序中所有的或部分的代码。因此，程序有可能进入了一个无限循环之中。请在 MPLAB IDE 中选择 Halt 重新获取控制。

6.3 错误消息

Cannot Find *filename* （不能找到 *filename*）

由于编译错误， *filename* 文件没有被创建。这就意味着将需要调试代码以便能顺利通过编译。

filename is missing （缺少 *filename* 文件）

请检查是否安装了缺少的文件。

Hex file not found （没有找到 Hex 文件）

项目未生成 hex 文件。代码中的错误将阻止生成 hex 文件，从而导致了该错误。

Language Tool Not Properly Installed （语言工具没有正确安装）

安装好新版本的 MPLAB IDE 后，请确认为 MPLAB IDE 提供了任何语言工具的正确位置。

No Debugging Information Available （没有可用的调试信息）

请确认已选择了在编译时创建 `.cod` 或 `.cof` 文件的选项。

Unresolved Breakpoints （不能解析的断点）

当断点未设置在有效代码行上，即断点设置在优化了的代码上或者设置在不具备与之相关的有效汇编指令的代码行上，就会显示该信息。欲知关于修正不能解析的断点的信息，请参阅**第 9.4 节 “Breakpoints 对话框”**。

6.4 限制

MPASM 汇编器限制

一些汇编器限制可以看作 MPLAB IDE 限制。这些限制包括：

- 调试 （COD）文件的文件名和文件路径名长度至多为 **62** 个字符。
- 汇编器不使用在 MPLAB IDE 中输入的头文件的路径信息。

欲知更多信息，请参阅 MPASM 汇编器文档中的 Limitations （限制）部分。

带 EEData 的 COD 文件

在 Disassembly （反汇编）窗口中可能不能正确显示带 EEData 的 COD 文件。

注：



第 2 部分——MPLAB IDE 使用参考

注：

第 7 章 MPLAB IDE 界面

7.1 简介

MPLAB IDE 界面是一个可以调整大小的窗口，它的操作独立于其余菜单项。此界面由菜单栏、工具栏、状态栏和任何打开的窗口和 / 或对话框组成。本章涉及以上各栏。而窗口和对话框将分别在第 8 章和第 9 章中进行讨论。

本章涉及以下主题：

- MPLAB IDE 菜单栏
- MPLAB IDE 工具栏
- MPLAB IDE 状态栏

7.2 MPLAB IDE 菜单栏

可以通过横跨界面顶部的菜单栏中的项来访问所有的 MPLAB IDE 功能。后跟省略号（...）的菜单项将打开一个对话框。

菜单项的旁边列出了菜单项的快捷 （热）键。例如：Print （打印）的快捷键是 Control-P （CTRL+P）。可在 *Configure>Settings* 中的 **Hot Keys** 选项卡下设置热键。

现有的菜单为：

- File
- Edit
- View
- Project
- Debugger
- Programmer
- Tools
- Configure
- Window
- Help

7.2.1 File

下面是 File 菜单中的菜单项。欲知更多有关编辑器和此处显示的项的信息，请参阅**第 11 章 “使用编辑器”**。

- New （新建）
  显示一个文件名为 “Untitled”（未命名）的空编辑器窗口。（参见**第 8.16 节 “File （编辑器）窗口”**。）在关闭窗口时，将提示您为文件命名。
- Open （打开）
  打开现有的源文件。可以通过在按下 CTRL 键或 Shift 键的同时单击文件名，在 “打开” 对话框（参见**第 9.8 节 “File Management 对话框”**）中选择多个文件。

- Close （关闭）
  关闭当前的编辑器窗口。如果文件自上一次保存之后已经被更改，会提示您保存更改。
- Save （保存）
  将当前的编辑器窗口保存到磁盘上它的原始文件名下。
- Save As （另存为）
  打开带有 "Encoding"（编码）列表框的 "Save As" 对话框 （参见**第 9.8 节 "File Management 对话框"**）。在 "Encoding" 列表框中选择文件编码的类型。Save As 命令允许将当前的编辑器窗口保存到磁盘中一个新的文件名下。
- Save All （保存所有）
  将所有打开的编辑器窗口保存到磁盘。
- Open Workspace （打开工作区）
  打开工作区会在打开一个新的工作区之前关闭前面的工作区。欲知更多有关工作区的信息，请参阅**第 4 章 "项目和工作区"**。
- Save Workspace （保存工作区）
  保存工作区将保存当前的工作区。可以通过在 <u>*Configure>Settings*</u> 中的 **Workspace** （工作区）选项卡下设置相关项以在关闭时自动保存工作区。
- Save Workspace As （将工作区另存为）
  打开 "另存为" 对话框 （参见**第 9.8 节 "File Management 对话框"**），允许在保存当前的工作区之前对它进行重命名 / 重定位。
- Close Workspace （关闭工作区）
  关闭工作区将返回默认的启动工作区配置。
- Import （导入）
  将调试文件或 hex 文件导入到 MPLAB IDE 项目。（参见**第 9.13 节 "Import 对话框"**。）
- Export （导出）
  从 MPLAB IDE 项目导出 hex 文件。（参见**第 9.6 节 "Export Hex File 对话框"**。）
- Print （打印）
  打印当前的编辑器窗口。将打开 "打印" 对话框，可在此设置打印机和打印选项。
- Recent Files （最近文件）
  显示在当前的 MPLAB IDE 会话过程中打开过的文件的列表。可在 <u>*Configure>Settings*</u> 的 **Workspace** 选项卡中设置所显示的文件数。
- Recent Workspaces （最近的工作区）
  显示在当前的 MPLAB IDE 会话过程中打开过的工作区的列表。可在 <u>*Configure>Settings*</u> 的 **Workspace** 选项卡中设置所显示的工作区数。
- Exit （退出）
  关闭 MPLAB IDE 应用程序。

7.2.2 Edit

下面是 Edit （编辑）菜单中的菜单项。欲知更多有关编辑器和此处显示的项的信息，请参阅**第 11 章 “使用编辑器”**。

- Undo （撤销）

  撤销对当前窗口所进行的上一次更改。当没有编辑操作可撤销时，此菜单命令将灰掉且不能选择。
- Redo （重复）

  恢复上一次 Undo 操作对当前窗口所进行的更改。当没有编辑操作可恢复时，此菜单命令将灰掉且不能选择。
- Cut （剪切）

  删除当前窗口中选定的文本并将它放在剪贴板上。在这个操作之后，可将所删除的文本粘贴到另一个 MPLAB 编辑器窗口、同一 MPLAB 编辑器窗口中的另一个位置或另一个 Windows 应用程序中。
- Copy （复制）

  将当前窗口中选定的文本复制到剪贴板上。在这个操作之后，可将所复制的文本粘贴到另一个 MPLAB 编辑器窗口、同一 MPLAB 编辑器窗口中的另一个位置或另一个 Windows 应用程序中。
- Paste （粘贴）

  将剪贴板上的内容粘贴到当前窗口的插入点上。仅可当剪贴板包含文本格式的数据时执行此操作。 **MPLAB 编辑器不支持位图粘贴或其他剪贴板格式**。
- Delete （删除）

  删除选定的文本。
- Select All （全选）

  选择 Edit 窗口中的所有文本和图形。
- Find （查找）

  打开 Find 对话框。（参见**第 9.11 节 “Find and Replace 对话框”**。）
- Find Next （查找下一个）

  查找 “查找文本” 的下一个实例。

  F3 重复上一次查找。

  Shift+F3 按上一次查找的相反方向查找。
- Replace （替换）

  打开 Replace 对话框。（参见**第 9.11 节 “Find and Replace 对话框”**。）
- Go to （转到指定行）

  转到编辑器窗口中指定的文本行。
- Advanced （高级）

  高级编辑功能。包括使选定的文本全部变为大写或小写、将文本或常规代码注释掉或者缩进文本或不缩进文本。

  除此之外还有一个 Match （匹配）功能。该功能可转到与光标处的括号相匹配的括号处。该功能适用于大括号、圆括号、尖括号和方括号的匹配查找。
- Bookmarks （书签）

  使用书签进行工作。切换书签 （交替的启用 / 禁止书签）、转到下一个或上一个书签或者禁止所有书签。
- Properties （属性）

  打开 Editor Options （编辑器选项）对话框 （参见**第 11.2.1 节 “Editor Options 对话框”**）。

7.2.3 View

下面是 View （视图）菜单中的菜单项。任何不能应用于选定器件的项都将被禁用。在此菜单中选择某项，就会使该项所对应的栏或窗口显示在桌面上。

- 第 **8.4** 节 **“项目窗口”**
- 第 **8.5** 节 **“Output 窗口”**
- 第 **7.3** 节 **“MPLAB IDE 工具栏”**
- 第 **8.6** 节 **“Disassembly Listing 窗口”**
- 第 **8.7** 节 **“Hardware Stack 窗口”**
- 第 **8.8** 节 **“Program Memory 窗口”**
- 第 **8.9** 节 **“File Registers 窗口”**
- 第 **8.10** 节 **“EEPROM 窗口”**
- 第 **8.11** 节 **“LCD Pixel 窗口”**
- 第 **8.12** 节 **“Watch 窗口”**
- 第 **8.13** 节 **“Special Function Registers 窗口”**
- *特定于工具的窗口*——取决于选定的工具 （*Debugger>Select Tool*），特定于工具的项 （例如，第 **8.14** 节 **“Trace Memory 窗口”**），也可能会出现在此菜单中。

7.2.4 Project

下面是 Project （项目）菜单中的菜单项。欲知更多有关项目的信息，请参阅**第 4 章 “项目和工作区”**。

- Project Wizard （项目菜单）

  使用 Project Wizard 可以帮助您建立新的项目。欲知更多有关此向导的信息，请参阅第 **4.2** 节 **“使用项目向导”**。
- New （新建）

  在工作区中创建新项目。

  打开 New Project 对话框。该对话框将要求输入新项目的名称和路径。（参见第 **9.14** 节 **“New Project 对话框”**。）
- Open （打开）

  将现有的项目添加到工作区并将它设置为活动项目。打开 Open Project 对话框（参见第 **9.8** 节 **“File Management 对话框”**。）
- Close （关闭）

  关闭工作区中当前的项目。会提示您在关闭之前保存当前的项目。

  关闭项目并不会关闭工作区。使用 *Edit>Workspace>Close* 关闭工作区。
- Set Active Project （设置活动项目）

  在工作区中选择一个项目作为活动项目。欲知更多有关活动项目的信息，请参阅第 **4.9** 节 **“在单个工作区中使用多个项目”**。

  要启用 Quickbuild，请选择 None。

- Quickbuild (*filename*)（快速编译 (文件名)）

  使用 MPASM 汇编器编译一个单独的汇编文件而不必创建项目 （无链接器）。必须在 Set Active Project （设置活动项目）下选择 None/Enable Quickbuild。必须在文件窗口中打开汇编文件并且该文件窗口必须是当前窗口。

  此过程可能会有汇编器限制。参见第 **6.4** 节 **"限制"**。
- Clean （清除）

  为活动项目删除所有的中间项目文件，诸如目标文件、 hex 文件和调试文件。当编译项目时，这些文件由其他文件重新创建。
- Build All （编译所有）

  通过编译 / 汇编所有文件来编译项目。必须有一个打开的项目才会显示此项。
- Make

  通过仅编译在上次编译后更改过的文件来编译项目。必须有一个打开的项目才会显示此项。
- Build Options （编译选项）

  使用 Build Options 对话框 （参见第 **9.5** 节 **"Build Options 对话框"**）设置和查看活动项目和单个文件的选项。
- Find in Project Files （在项目文件中查找）

  在当前项目的多个文件中查找文本。（参见第 **9.10** 节 **"Find In Project Files 对话框"**。）在 Output 窗口中显示查找结果。
- Save Project （保存项目）

  保存当前项目。
- Save Project As （将项目另存为）

  打开 Save Project As 对话框。（参见第 **9.18** 节 **"Save Project As 对话框"**。）
- Add Files to Project （将文件添加到项目）

  将文件插入到当前项目中。（参见第 **9.8** 节 **"File Management 对话框"**。） MPLAB IDE 将根据文件的类型将文件归类到项目窗口树中正确的类型。
- Remove Files from Project （从项目中移除文件）

  将文件从当前项目中移除 （删除）。而不是将文件从目录中删除。
- Select Language Toolsuite （选择语言工具包）

  选择将用于项目的语言工具包，例如， Microchip Toolsuite。（参见第 **9.20** 节 **"Select Language Toolsuite 对话框"**。）
- Set Language Tool Locations （设置语言工具的路径）

  为将要在项目中使用的语言工具设置路径 / 目录，即将 MPLAB IDE 中的语言工具名称 （如 MPASM 汇编器）与一可执行文件 （如 `C:\Program Files\MPLAB IDE\MCHIP_Tools\ mpasmwin.exe`）匹配。（参见第 **9.21** 节 **"Set Language Tool Location 对话框"**。）
- Select Version-Control System （选择版本控制系统）

  设置项目使用来自版本控制系统 （Visual Source Safe）的文件。欲知更多信息，请参阅第 **4.6** 节 **"使用版本控制系统 （VCS）"**。

7.2.5 Debugger

下面是 Debugger 菜单中的菜单项。

> **注：** 在使用调试器的时候，如果选定的器件有 EEPROM 数据，并且 （1）已启用了编程器或 （2）打开了 EEPROM 窗口，由于以上任一情况都会在每步尝试访问数据，单步运行将变得非常缓慢。为了提高速度，请禁止编程器或关闭 / 最小化 EEPROM 窗口。

- Select Tool （选择工具）

  选择一个调试工具。默认选择为 None。可用的调试器列表根据已安装的调试工具而定。列表中条目的顺序与安装顺序一致。
- Clear Memory （清空存储器）

  清空全部或仅清空在此项目中使用的特定类型的 MPLAB IDE 存储器，例如程序、数据、 EEPROM 和配置。
- *基本调试选项*
- *工具特定的选项*——取决于选定的调试工具 （*Debugger>Select Tool*），其他工具特定的项 （如，“Stopwatch”），也可能会出现在此菜单中。

基本调试选项

一旦选择了调试工具， Debugger 菜单就会添加以下选项：

- Run

  执行程序代码直到遇到断点或者选择了 “Halt”。

  程序从当前的程序计数器开始执行 （如状态栏中所示）。当前程序计数器的位置也可以由 Program Memory 窗口中的一个箭头表示。在运行程序时，将禁止几个其他功能。
- Animate

  单步连续运行使得调试器在运行程序时实际执行单步运行，在运行时会更新寄存器的值。

  Animate 比 Run 功能运行要慢，但是这样做允许在 Special Function Register 窗口或 Watch 窗口中查看寄存器值的变化过程。

  要暂停单步连续运行，请使用菜单选项 *Debugger>Halt* 而不是工具栏上的 Halt 或快捷键 **F5**。
- Halt

  暂停 （停止）程序代码的执行。当单击 Halt 时，将更新状态信息。
- Step Into

  单步运行整个程序代码。

  对于汇编代码来说，此命令执行一条指令 （单周期或多周期指令），然后暂停。在执行了一条指令之后，所有窗口都被更新。

  对于 C 代码来说，此命令执行一行 C 代码，这可能意味着执行一条或多条汇编指令，然后停止。在执行完之后，所有窗口被更新。

  > **注：** 不要单步跳入 `Sleep` 指令。

- Step Over

  在当前程序计数器处执行指令。当遇到 CALL 指令时，单步跳过调用的子程序并在 CALL 之后的地址处暂停。如果单步跳过太长或者显示出已经 “挂起”，单击 Halt。
- Step Out

  单步跳出子程序。如果正在单步运行子程序代码，可以使用 Step Out 在完成执行子程序的剩余部分后在 CALL 之后的地址处暂停。
- Reset

  执行指定的复位，可以是 MCLR、 Watchdog Timer （看门狗定时器）、 Brown Out （欠压）或 Processor （处理器）复位。 Reset 选项和操作取决于选定的器件。
- Breakpoints

  打开 Breakpoint 对话框。在此对话框中设置多个断点。欲知设置断点的其他方法，请参阅**第 3.18 节 “使用断点”**。
- Settings

  打开特定于工具的设置窗口。在此设置工具功能。并查找工具限制。

7.2.6 Programmer

下面是 Programmer 菜单中的菜单项。

- Select Programmer （选择编程器）

  选择一个编程器。默认选择为 None。可用的编程器列表取决于已安装的调试工具。列表中条目的顺序与安装顺序一致。
- *基本编程器选项*
- *特定于编程器的选项*——取决于选定的编程器 （*Programmer>Select Programmer*），其他特定于编程器的项，如，“Load SQTP File”（装载 SQTP 文件），也可能会出现在此菜单中。

基本编程器选项

根据所选择的编程器，出现在 Programmer 菜单上的选项可能有所不同。一般出现的基本项有：

- Enable Programmer （启用编程器）

  在 MPLAB IDE 和编程器之间建立通信。如果该编程器已经被启用，此选项会灰掉。
- Disable Programmer （禁止编程器）

  结束 MPLAB IDE 和编程器之间的通信。如果该编程器已经被禁止，此选项会灰掉。
- Program （编程）

  编程指定的存储区：程序存储器、配置位、 ID 单元和 / 或 EEPROM 数据。
- Verify （校验）

  校验对指定的存储区的编程，这些区包括：程序存储器、配置位、 ID 单元和 / 或 EEPROM 数据。
- Read （读）

  读指定的存储区：程序存储器、配置位、 ID 单元和 / 或 EEPROM 数据。
- Blank Check All （完全空白检查）

  检查所有器件存储区是否被擦除 / 为空白。

- Blank Check OTP （OTP 空白检查）
  检查 OTP 器件，查看程序存储器、数据存储器和 EEPROM 存储器是否被擦除 / 为空白。
- Erase Flash Device （擦除闪存器件）
  擦除 PICmicro 闪存 MCU 器件上的所有数据，包括存储器、 ID 和配置位。
- Reset Program Statistics （复位编程统计）
  将编程统计 （例如错误数）设置为默认值。
- Download OS （下载 OS）
  为编程器下载最新的操作系统。

注：	必须升级了 PICSTART Plus 之后才可使用此功能。请遵循该工具的帮助中的说明。

- About （关于）
  在此对话框中查看与工具相关的信息。
- Settings
  打开工具特定的设置对话框。在此对话框中设置与工具相关的信息，即 Memory Ranges （存储区范围）和 Communications Port Setup （通信端口设置），如果可用的话，还可设置 Voltages （电压）和 SQTP。

7.2.7 Tools

下面是 Tools 菜单中的菜单项。

- MPLAB Macros （MPLAB 宏）
  将 Microsoft 宏功能和 MPLAB IDE 结合使用。
- Visual Initializer （可视化初始化程序）
  可视地设置开发代码。欲知更多有关此工具如何操作的信息，请参阅 MPLAB VDI 的在线帮助。

7.2.8 Configure

下面是 Configure 菜单中的菜单项。

注：	根据选定的器件和调试工具，可能并不是所有项都可用。

- Select Device
  为开发模式选择器件。（参见第 **9.19** 节 **"Select Device 对话框"**。）在 Debugger 或 Programmer 菜单下选择开发工具。
- Configuration Bits
  选择器件配置位的值。（参见第 **8.15** 节 **"Configuration Bits 窗口"**。）设置这些值将影响调试器和编程器的操作。
- External Memory
  选择是否使用外部存储器。同时也指定外部存储器的范围。（参见第 **9.7** 节 **"External Memory Setting 对话框"**。）
- ID Memory （ID 存储区）
  输入值到 ID 存储区。（参见第 **9.24** 节 **"User ID Memory 对话框"**。）
- Settings
  输入工作区、调试器、程序装载、热键和项目的默认设置。（参见第 **9.22** 节 **"Settings 对话框"**。）

7.2.9 Window

下面是 Window 菜单中的菜单项。

- Close All （关闭所有）
  关闭所有打开的窗口。
- Cascade （层叠）
  层叠排列打开的窗口以便看到每个标题栏。
- Tile Horizontally （水平平铺）
  一个接一个的用较小的尺寸水平排列打开的窗口。
- Tile Vertically （垂直平铺）
  一个接一个的用较小的尺寸垂直排列打开的窗口。
- Arrange Icons （排列图标）
  在 IDE 的底部排列所有已最小化为图标的窗口。
- *Open windows （打开窗口）*
  显示所有打开的窗口的列表。单击一个窗口名称，让该窗口变为当前窗口。

7.2.10 Help

下面是 Help 菜单中的菜单项。

- Topics
  从此对话框的列表中选择帮助文件。（参见第 **9.12 节 “Help Topics 对话框”**。）
- Readme Files （自述文件）
  查看 Microchip 工具所有可用的 Readme 文件的 HTML 列表。单击链接查看实际的文件。
- About MPLAB IDE （关于 MPLAB IDE）
  查看 MPLAB IDE 商标和组件的版本信息。（参见第 **9.2 节 “About MPLAB IDE 对话框”**。）

7.3 MPLAB IDE 工具栏

根据正在使用的功能或工具， MPLAB IDE 会显示不同的工具栏。工具栏中的图标提供执行常规任务的快捷方式。

工具栏功能

- 单击并拖动工具栏可以让它变成浮动的工具栏。
- 单击并将工具栏拖动到 MPLAB IDE 界面的顶部或者边上来停驻它。
- 单击并将工具栏拖出 MPLAB IDE 界面。
- 将鼠标悬停在图标上就会弹出图标名称。
- 右击工具栏，更改内容或显示 / 隐藏工具栏。

可用的工具栏

- 标准工具栏
- 项目管理器工具栏
- 调试工具栏
- 编程器工具栏
- 校验和工具栏

7.3.1 标准工具栏

标准 （编辑）工具栏目前包含具有以下功能的按钮图标：

- New File——打开新建文件窗口
- Open File——在窗口中打开现有的文件
- Save File——将当前文件窗口的内容保存到一个文件
- Cut——将选定的文本剪切到剪贴板
- Copy——将选定的文本复制到剪贴板
- Paste——粘贴来自剪贴板的文本
- Print File——打印当前文件窗口中的内容
- Find——打开 Find 对话框，在当前文件窗口中查找文本
- Help——显示 MPLAB IDE Help 选择对话框

7.3.2 项目管理器工具栏

项目管理器工具栏目前包含具有以下功能的按钮图标：

- New Project——设置新项目的名称和位置
- Open Project——打开现有的项目
- Save Workspace——将当前项目和工作区保存为文件
- Build Options——查看或更改项目设置
- Find in Project Files——打开 Find 对话框，在所有项目文件中搜索文本

如果已经装载了项目，还可能找到其他项：

- Make——仅编译当前项目中已经更改的文件
- Build All——编译当前项目中的所有文件

7.3.3 调试工具栏

调试工具栏目前包含具有以下功能的按钮图标：

- Run——运行程序
- Halt——暂停程序执行
- Animate——连续单步运行指令，使用 *Debugger>Halt* 暂停执行
- Step Into——单步跳入下一条指令
- Step Over——单步跳过下一条指令
- Step Out——单步跳出子程序
- Reset——执行 MCLR 复位

根据选定的调试工具，还可能出现其他图标。

7.3.4 编程器工具栏

根据所选择的编程器，编程器工具栏上出现的按钮图标也会有所不同。通常出现的基本图标有：

- Blank Check All/Blank Check （完全空白检查 / 空白检查）——检查器件存储器是否为空。
- Read——读取在 *Programmer>Settings* 的 **Program** 选项卡中指定的器件存储区。
- Program——为在 *Programmer>Settings* 的 **Program** 选项卡中指定的器件存储区编程。
- Verify——验证目标存储器是否已经被正确编程。
- Erase Flash Device （擦除闪存器件）——如果器件具有闪存程序存储器，则将其擦除。
- Program Statistics （编程统计）——显示编程统计数据，如编程成功、失败的次数以及尝试编程的总次数。

7.3.5 校验和工具栏

此工具栏仅显示校验和的值。可以在该器件的编程规范中找到校验和的算法。可以在我们的网站上找到编程规范。

当器件有代码保护时，可以使用未受保护的校验和来确定器件的校验和。

以前，此项在状态栏上。现在它在状态栏中的位置被存储区信息占用了。

7.4 MPLAB IDE 状态栏

状态栏提供与 MPLAB IDE 会话状态相关的最新信息。

在运行应用程序时，它会显示 “Running”和一个进程条。

当不运行应用程序时，它所提供的信息包括：

编号	标题	典型项	说明
1	当前调试工具	MPLAB SIM	显示当前选定的调试工具
2	当前编程器	PICSTART Plus	显示当前选定的编程器
3	当前处理器	PIC18F452	显示当前选定的处理器
4	当前程序计数器	pc:0x5f	显示当前程序计数器
5	w 寄存器的当前值	W:0x00	显示 w 寄存器的当前值。
6	状态位	ov Z dc c	大写为置位 （1） 小写为复位 （0）
7	全局断点使能	Bk On	显示全局断点使能的当前状态。
8	处理器频率	4MHz	显示处理器的当前频率。
9	存储信息	bank 0	显示数据存储器中的当前存储区
10*	打开窗口中的行号和列号	Ln 1 Col 1	显示文件中的当前行号和列号
11*	插入 / 改写	INS	在插入和改写之间切换输入模式。 INS = 插入字符 OVR = 改写字符
12*	写 / 只读	WR	显示写 / 只读状态 WR = 可编辑文件 RO = 只读文件

* 只有在文件 （编辑器）窗口为当前窗口时才可用。

注：



第 8 章 MPLAB IDE 窗口

8.1 简介

MPLAB IDE 窗口的操作和普通的 Windows 应用程序相同。它具备标准的 Windows 功能部件，包括窗口大小调整按钮以及垂直和水平的滚动条。除此之外，MPLAB IDE 窗口还具有以下特有的功能：

- 就地编辑数据
- 可停驻 （Dockable）窗口
- 单击并拖动可以改变窗口中列的宽度和顺序
- 右击 （上下文）菜单可以更改窗口的字体和颜色

在 View 菜单下列出了大多数可在 MPLAB IDE 中使用的窗口。根据已安装的工具，可能还有工具特定的窗口可供查看。

常规

- 更改窗口数据和属性
- 代码显示窗口中的符号

View 菜单

- 项目窗口
- Output 窗口
- Disassembly Listing 窗口
- Hardware Stack 窗口
- Program Memory 窗口
- File Registers 窗口
- EEPROM 窗口
- LCD Pixel 窗口
- Watch 窗口
- Special Function Registers 窗口
- Trace Memory 窗口

Configure 菜单

- Configuration Bits 窗口

File 菜单

- File （编辑器）窗口

8.2 更改窗口数据和属性

根据窗口的类型和用途， MPLAB IDE 窗口具有部分或全部所列的属性。

8.2.1 更改窗口数据——就地编辑或从列表中选择

可以像下面描述的那样编辑 MPLAB IDE 窗口中的数据。若不能编辑数据，则表示不允许您更改此信息。

- 可以 “就地” 编辑数据，即，双击选择一项，然后输入一个新值。
- 若数据只可能为特定的几个值，则可从下拉菜单中选择一个值作为该数据的值。

8.2.2 更改窗口位置——停驻 / 还原

停驻

- 通过从窗口系统菜单 （单击左上角）中选择 “Dockable”（可停驻）使窗口可停驻 （能够附着在 MPLAB IDE 桌面的顶部或四周）。

> **注：** 文件 （编辑器）窗口不能停驻。

- 通过将可停驻窗口拖动到所需停驻的位置即可停驻它们。
- 可停驻窗口可被悬空，即被拖出 MPLAB IDE 桌面。

不可停驻

- 通过双击夹条 可使已停驻的窗口还原。
- 通过右击夹条并从菜单中选择 “Dockable”可使已停驻的窗口还原。
- 通过右击标题栏并从菜单中选择 “Dockable”可使未停驻的窗口不可停驻。

8.2.3 更改窗口中的列——大小、可见性和顺序

- 使用以下方法可重新调整列的大小：将光标移到列之间的线条上直到它变为调整大小光标为止，然后单击并拖动以使列变宽或变窄。
- 使用以下方法可使列可见 / 不可见：
  - 右击任一列的标题并从列表中选取一项，即可使该项所对应的列可见。
  - 右击任一列的标题并选择 More （更多），打开窗口的 Properties 对话框，可在该对话框的 **Column Settings** （列设置）选项卡 （参见第 **9.17** 节 **“Properties 对话框”**）上设置列的可见性。
  - 在窗口中右击并选择 Properties。可在 **Column Settings** 选项卡上设置列的可见性。
- 使用以下方法可为列排序：
  - 单击任一列的标题并将其拖到所需的位置。
  - 右击任一列的标题并选择 More，打开窗口 Properties 对话框，可在该对话框的 **Column Settings** 选项卡上 （参见第 **9.17** 节 **“Properties 对话框”**）设置列的位置。
  - 在窗口中右击并选择 Properties。可在 **Column Settings** 选项卡设置列的位置。

8.2.4 更改窗口的字体和颜色

- 通过在窗口中右击并选择 Properties，可在 **General** 选项卡 （参见第 **9.17** 节 **“Properties 对话框”**）上指定窗口的字体并改变窗口的颜色。
- 通过在窗口中右击并选择 Properties，可在 **Background Colors** （背景颜色）选项卡 （参见第 **8.13** 节 **“Special Function Registers 窗口”**）中选择 SFR 窗口的背景颜色。

8.3 代码显示窗口中的符号

在一些窗口左侧的灰色区域中，可能出现下列符号。

符号	说明
(绿色实心箭头)	程序计数器的当前位置 （代码行）。程序执行暂停 （停止）。
(绿色空心箭头)	运行程序前程序计数器的位置。程序目前正在执行 （运行）。
(红色 B)	断点设置。
(红色空心八边形)	断点禁用。
(黄色 B)	在地址上设置断点。由于一行代码可能对应多个地址，因此仅在文件 （编辑器）窗口中显示该符号。更多信息，请参见第 **3.18** 节 **“使用断点”**。
(对勾)	在此位置启用代码覆盖 （仅在选用 MPLAB ICE 2000/4000 的情况下，出现在程序存储器窗口中）。
(脉冲)	在此位置设置复杂触发点 （仅在选用 MPLAB ICE 2000/4000 的情况下，出现在程序存储器窗口中）。
(空方框)	选入跟踪 （文件窗口）。更多信息，请参见第 **11.5.1** 节 **“筛选跟踪”**。
(带叉方框)	排除跟踪 （文件窗口）。更多信息，请参见第 **11.5.1** 节 **“筛选跟踪”**。

8.4 项目窗口

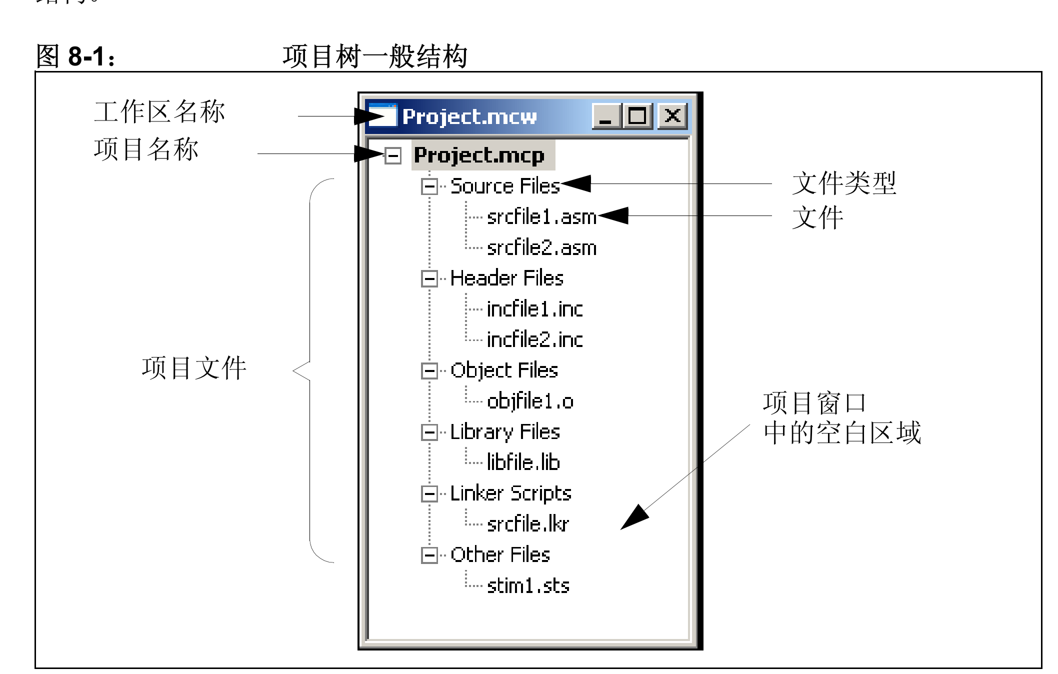
项目窗口包含项目的摘要信息。选择 *View>Project* 可查看或关闭项目窗口。

- 项目窗口显示
- 项目窗口菜单
- 项目窗口 FAQ

8.4.1 项目窗口显示

项目窗口的标题栏包含项目工作区的名称。该窗口本身包含一个列表，该列表包括项目和按照文件类型以树形排列的项目文件。图 8-1 中所示为一个项目的项目树的一般结构。

图 8-1： 项目树一般结构



如果项目名称为黑体，则表示其为活动项目。如果项目名称后面跟着一个星号（例如 `c:\project1\project.mcp*`），则表示该项目的设置已更改，或者该项目为一个新的项目，需要保存。

有关在此窗口中列出的文件类型的信息，请参见**第 4.5 节 “项目文件夹和文件”**。

8.4.2 项目窗口菜单

根据在项目窗口中单击的位置，可以得到不同的菜单。

- 显示菜单
- 项目菜单
- 文件类型菜单
- 文件菜单

8.4.2.1 显示菜单

如果在项目窗口中右击任何空白的区域，就将显示带有项目显示命令的菜单。

- Refresh （刷新）

  此命令使 IDE 暂停当前任务以检查所有项目中的所有文件的状态。

  对于版本控制系统文件，会在项目窗口中更新这些文件的状态。

  对于已重新定位的丢失文件，“file not found”（找不到文件）文本将被去除。

- Preferences （首选项）

  打开对话框以设置项目首选项。参见**第 9.15 节 “Project-Display Preferences 对话框”**。

8.4.2.2 项目菜单

如果右击一个项目文件，将显示一个带有项目命令的菜单。

- Set Active （激活）

  在工作区中选择一个项目作为活动项目。有关活动项目的更多信息，请参见**第 4.9 节 “在单个工作区中使用多个项目”**。要启用快速编译，请选择 None （无）。

- Clean

  删除所有的中间项目文件，如目标文件、 hex 文件和调试文件。编译项目时，这些文件由其他文件重新创建。

- Build All

  要让此选项可见，必须先打开一个项目。

  编译该项目中的所有文件，即编译 / 汇编项目中的所有文件。在项目窗口中的右键菜单中也有 Build All 命令。

- Make

  要让此选项可见，必须先打开一个项目。

  只编译该项目中已更改的文件，即编译 / 汇编在项目中指定的这些文件。在项目窗口中的右键按钮菜单中也有 Make 命令。

- Build Options

  项目和各个文件的设置和查看选项。参见**第 9.5 节 “Build Options 对话框”**。

- Save

  保存活动项目。

- Save As

  将活动项目保存到新的位置 / 名称。参见**第 9.18 节 “Save Project As 对话框”**。

- Close

  在工作区中关闭 / 删除选定的项目。

- Add Files

  将文件插入到项目中。 MPLAB IDE 将根据文件的类型将其归类到项目窗口树中正确的类型中。

  > **注：** 将头文件添加到项目窗口中并不会把这些文件添加到项目编译中。必须在代码中使用 `#include` 语句，才能在编译时调用这些文件。

- Reload （重载）

  此命令使 IDE 暂停当前任务并从磁盘中再次读入项目文件。随后将立即执行 Refresh 命令。如果修改了项目，但随后又决定取消这些修改，该功能非常有用。只要尚未保存和改写旧的项目，就可以使用 “Reload”重载并恢复原来的项目。
- Refresh

  这个命令使 IDE 检查项目中所有文件的状态。

  对于版本控制系统文件，会在项目窗口中更新这些文件的状态。

  对于已重新定位的丢失文件，“file not found”文本将被去除。
- Select Language Toolsuite

  选择将为您的项目使用的工具包，例如， Microchip MPASM 工具包。参见**第 9.20 节 “Select Language Toolsuite 对话框”**。
- Select Version-Control System

  设置项目以使用来自版本控制系统的文件。参见**第 9.25 节 “Version-Control 对话框”**。

8.4.2.3 文件类型菜单

如果在项目树中右击一种文件类型，将显示带有文件类型命令的菜单。

- Library Link Order （库链接顺序）（仅针对库文件）

  打开 Library Link Order 对话框，它会显示库文件的实际顺序并允许将它们重新排序。这对于 MPLAB C30 用户很重要。
- Add Files

  将文件插入到项目中。 MPLAB IDE 将根据文件的类型将文件归类到项目窗口树中正确的类型。

注：	将头文件添加到项目窗口中并不会把这些文件添加到项目编译中。必须在代码中使用 `#include` 语句，才能在编译时调用这些文件。

- Filter （筛选）

  改变文件筛选的方式，以确定要使用的文件类型。例如，要能将具有 .a 扩展名的汇编文件添加到 Source Files 下，必须将 .a 添加到 Filter 对话框的文件扩展名列表中。

8.4.2.4 文件菜单

如果在项目树中右击一个文件，将显示带有文件命令的菜单。

注：	如果正在使用版本控制系统，则可对选定文件使用更多的命令。

- Assemble/Compile

  必要时汇编 / 编译选定的文件。
- Build Options

  项目和各个文件的设置和查看选项。参见**第 9.5 节 “Build Options 对话框”**。
- Edit

  必要时在窗口中打开选定的文件以进行编辑。
- Remove

  从项目中删除选定的文件。并不是将该文件从目录中删除。

8.4.3 项目窗口 FAQ

如何：

- 创建 / 更新项目？
  参见第 **4.3** 节 **"创建 / 更新任何项目"**。
- 编译项目？
  在工作区中右击想要编译的项目。在菜单中选择 Build All 来编译整个项目；或者选择 Make 来重新编译 / 重新汇编任何已更改的文件，然后再次编译项目。
  即使选定的项目不是活动的项目，仍会将此次编译的结果装入程序存储器。
- 在工作区中使用项目？
  参见第 **4** 章 **"项目和工作区"**。
- 激活项目？
  要将项目激活，请选择 *Project>Set Active Project>projectname.mcp*，其中 `projectname.mcp` 是项目的名称，或者在项目窗口中右击该项目并选择 Set As Active Project。

8.5 OUTPUT 窗口

选择 *View>Output*，打开 Output 窗口。此窗口包含了有关程序输出的信息。

- **Build** （编译）标签——列出了来自项目编译的消息。编译消息与选定的语言工具（*Project>Select Language Toolsuite*）和设定的编译选项 （*Project>Build Options>Project* 有关。
- **Version Control** 标签——当在项目中使用版本控制系统 （*Project>Version Control*）时显示版本控制信息。
- **Find in Files** 标签——列出了 *Project>Find in Project Files* 的结果。
- 根据选定的功能，可能还会有其他的输出标签。

以下是在 Output 窗口右键菜单中的菜单项。

- Select All
  在 Edit 窗口中选择所有文本和图形。
- Copy
  将当前窗口中选定的文本复制到剪贴板。随后，您可将所复制的文本粘贴到另一个 MPLAB 编辑器窗口、同一个 MPLAB 编辑器窗口中的另一个位置或另一个 Windows 应用程序。
- Clear Page （清除页面）
  删除选定输出标签中的所有文本。

8.6 DISASSEMBLY LISTING 窗口

选择 *View>Disassembly Listing* 在该窗口中查看反汇编代码。可以在代码中设置断点。代码执行信息可以符号的形式显示在窗口左侧的灰色区域中。

> 注： 包含 EEData 的 COD 文件可能无法正确执行。在行号表中程序和 EEData 的起始位置都显示为 Line 0 （第 0 行）。

以下是 Disassembly 窗口右键菜单中的菜单项。

- Set/Remove Breakpoint （设置 / 删除断点）
  在当前选定的行上设置或删除断点。
- Enable/Disable Break （启用 / 禁用断点）
  在当前选定的行上启用或禁用断点。
- Breakpoints （断点）
  禁用、启用或删除所有断点。
- Run To Cursor （运行到光标）
  将程序运行到当前的光标位置。在以前的版本中称为 Run to Here （运行到此）。
- Set PC at Cursor （将 PC 设置在光标处）
  将程序计数器 （PC）设置到光标位置。
- Copy
  将选定文本复制到剪贴板。可通过两种方法选择文本：（1）单击并拖动鼠标，以覆盖要选择的文本；（2）单击文本的起始位置，然后按住 shift 键并同时单击文本末尾。
- Select All
  选择窗口中的所有文本。
- Output to File （输出到文件）
  使用 "另存为"对话框将在窗口中显示的内容写入一个文本文件 （参见第 **9.8** 节 **"File Management** 对话框"）。
- Print
  打印该窗口的内容。
- Properties
  打开 Disassembly Options （反汇编选项）对话框。设置显示和功能选项。此对话框与 Editor Options （编辑器选项）对话框类似 （参见第 **11.2.1** 节 **"Editor Options** 对话框"）。

8.7 HARDWARE STACK 窗口

Hardware Stack 窗口显示硬件堆栈的内容。可用的堆栈深度取决于选定的器件。

> 注： 如果清除了 Disable Stack Overflow Warning （禁止堆栈溢出警告），当发生堆栈上溢或下溢时 MPLAB IDE 将显示警告。并非所有的处理器模块都支持此功能。

- Hardware Stack 窗口显示
- Hardware Stack 窗口菜单
- Hardware Stack 窗口 FAQ

8.7.1 Hardware Stack 窗口显示

以下各列用于显示数据：

- TOS （如果存在）——指示栈顶 （top of stack， TOS）的指针。
- Stack Level （堆栈深度）——堆栈的深度。最大的堆栈深度取决于选定的器件。
- Stack Return Address——堆栈中的返回地址。
- Location （位置）——函数名 + 偏移量。有关函数在堆栈中的位置的信息。

Stack Return Address

显示硬件堆栈的当前内容。仍会显示以前用过的值，因为这些值仍在器件中。栈顶（TOS）指示器显示堆栈指针的当前位置。

8.7.2 Hardware Stack 窗口菜单

以下是 Hardware Stack 窗口右键菜单中的菜单项。

- Close
  关闭此窗口。
- Pop Stack （弹出堆栈）
  将栈顶中的地址弹出到程序计数器中。
- Set Top of Stack （设置栈顶）
  将当前光标位置设为栈顶。
- Center Stack Location （居中堆栈位置）
  将当前选定的行置于窗口的中央。
- Output to File
  将在窗口中显示的内容写入一个文本文件。使用 “另存为”对话框 （参见第 **9.8 节 “File Management 对话框”**），还要在该对话框的 “Output Range”（输出范围）中输入将要输出的内容的 “Start”（起始）和 “End”（结束）行号。
- Print
  打印此窗口的内容。
- Refresh
  刷新此窗口中的数据。
- Properties
  设置字体和颜色。参见第 **9.17 节 “Properties 对话框”**。

8.7.3 Hardware Stack 窗口 FAQ

如何：

- 理解 Return Address 标签下的信息？
  参见第 **8.7.1 节 “Hardware Stack 窗口显示”**。
- 弹出堆栈？
  在窗口中右击以打开一个菜单。选择 Pop Stack。
- 设置栈顶 （TOS）？
  在窗口中右击以打开一个菜单。选择 Set Top of Stack。

8.8 PROGRAM MEMORY 窗口

Program Memory 窗口为当前选定的处理器显示程序存储器范围内的存储单元。如果选定器件支持并启用了外部程序存储器，则外部程序存储器也将出现在 Program Memory 窗口中。

- Program Memory 窗口显示
- Program Memory 窗口菜单
- Program Memory 窗口 FAQ

8.8.1 Program Memory 窗口显示

通过单击程序存储器窗口底部的几个按钮可以改变窗口中操作码的显示方式：

- Opcode Hex （十六进制操作码）
- Machine 或 Symbolic
- PSV Mixed （PSV 混合）（仅适用于 dsPIC 器件）
- PSV Data （PSV 数据）（仅适用于 dsPIC 器件）

8.8.1.1 OPCODE HEX （十六进制操作码）

此格式将程序存储器信息显示为十六进制码。该窗口将具有以下列：

- Address——下一列中操作码的十六进制地址。
- *Opcode Blocks* （操作码块）——以 2 个字节或 3 个字节的块显示的十六进制操作码。对于大多数 PICmicro MCU 的程序存储器而言，每块代表一个字。对于 PIC18CXXX 器件，每块代表 2 个字节。对于 dsPIC 器件，每块代表 3 个字节。
  突出显示的操作码块代表程序计数器的当前位置。
- ASCII——操作码的对应行的 ASCII 码表示。

8.8.1.2 MACHINE 或 SYMBOLIC

Machine 格式显示不带符号信息的反汇编十六进制码。 Symbolic 格式显示带符号的反汇编十六进制码。该窗口将具有以下列：

- *Debug Info* （调试信息）——对调试有用的信息。箭头显示程序计数器的当前位置。
- Line——参考行号
- Address——操作码的十六进制地址。
- Opcode——以 2 个字节或 3 个字节的块显示的十六进制操作码。对于大多数 PICmicro MCU 的程序存储器而言，每块代表一个字。对于 PIC18CXXX 器件，每块代表 2 个字节。对于 dsPIC 器件，每块代表 3 个字节。
- Label （标号）（仅以符号表示）——符号格式的操作码标号。
- Disassembly——操作码助记符的反汇编形式。

8.8.1.3 PSV MIXED （PSV 混合）（仅适用于 dsPIC 器件）

此格式显示程序存储器的操作码和 PSV 区域 （CORCON 寄存器的 PSV 位置 1）。该窗口将具有以下列：

- *Debug Info*——对调试有用的信息。箭头显示程序计数器的当前位置。
- Line——参考行号。
- Address——操作码的十六进制地址。
- Opcode——十六进制操作码，以 3 字节的块显示。
- PSV Address （PSV 地址）——操作码在数据空间内的十六进制地址。
- Data——以数据格式显示的操作码。
- Label——符号格式的操作码标号。
- Disassembly——操作码助记符的反汇编形式。

欲知更多有关 dsPIC 器件的信息，请参阅 《*dsPIC30F 系列参考手册*》（DS70046C_CN）。

8.8.1.4 PSV DATA （PSV 数据）（仅适用于 dsPIC 器件）

当程序空间在数据空间中可见时 （CORCON 寄存器中的 PSV 位置 1），此格式将程序存储器显示为文件寄存器。该窗口将具有以下列：

- Address——数据的程序空间十六进制地址。
- PSV Address——数据的数据空间十六进制地址。
- Data Blocks （数据块）——十六进制数据，以 3 字节的块显示。

突出显示的数据块代表程序计数器的当前位置。

- ASCII——数据对应行的 ASCII 表示。

欲知更多有关 dsPIC 器件的信息，请参阅 《dsPIC30F 系列参考手册》（DS70046C_CN）。

8.8.2 Program Memory 窗口菜单

以下是 Program Memory 窗口右键菜单中的菜单项。

- Close
  关闭此窗口。
- Set/Remove Breakpoint （仅机器码 / 符号格式）
  在当前选定的行上设置或删除断点。
- Enable/Disable Break （仅机器码 / 符号格式）
  在当前选定的行上启用或禁用断点。
- Breakpoints
  禁用、启用或删除所有断点。
- Run To Cursor （运行到光标）
  将程序运行到当前的光标位置。在以前的版本中称为 Run to Here （运行到此）。
- Set PC at Cursor （将 PC 设置在光标处）
  将程序计数器 （PC）设置到光标位置。
- Center Debug Location （居中调试位置）
  将当前的 PC 行置于窗口的中央。
- Cursor Tracks Debug Location （光标跟踪调试位置）
  光标 （箭头）将跟踪当前的调试位置。

- Find

  在此窗口中查找在 Find 对话框中指定的文本。
- Find Next （查找下一个）

  查找 “查找文本” 的下一个实例。

  F3 重复上一次查找。

  Shift+F3 按上一次查找的相反方向查找。
- Go To （转到）

  转到在 Go To 对话框中指定的地址。
- Import Table （导入表）

  打开 Import （导入）对话框 （参见第 **9.8** 节 **“File Management** 对话框**”**）。要使此菜单项可用，必须选择 “Opcode Hex”。
- Export Table （导出表）

  打开 Export As （导出为）对话框 （参见第 **9.8** 节 **“File Management** 对话框**”**）。要使此选项可用，必须选择 “Opcode Hex”（十六进制操作码）。
- Fill Memory （填充存储器）

  用 Data （数据）框中的值填充从 Start Address （起始地址）到 End Address （结束地址）范围内的存储区。参见第 **9.9** 节 **“Fill Memory/Registers** 对话框**”**。
- Output to File

  将在窗口中显示的内容写入一个文本文件。使用 “另存为” 对话框 （参见第 **9.8** 节 **“File Management** 对话框**”**），此外还要在该对话框的 “Output Range” 中选择范围的类型，“Lines” 或 “Address”，然后输入 “Start” 和 “End” 的值，以指定输出范围。
- Print

  打印此窗口的内容。
- Refresh

  刷新此窗口中的数据。
- Properties

  设置此窗口的属性。参见第 **9.17** 节 **“Properties** 对话框**”**。

8.8.3 Program Memory 窗口 FAQ

如何：

- 为支持外部 （片外）存储器的器件启用外部存储器？

  请选择 *Configure>External Memory*。在 External Memory 对话框中，选取 “Use External Memory”（使用外部存储器），输入一个范围，然后单击 **OK**。现在 External Memory 将显示在 Program Memory 窗口中。

  > **注：** 对于某些工具，可能必须先主动地将外部存储器上载到 MPLAB IDE，才能在该窗口中显示相应的值。

- 用值填充程序存储器？

  在该窗口中右击并选择 “Fill Memory” 以打开一个可以在其中输入填充数据的对话框。
- 设置断点？

  参见第 **3.18** 节 **“使用断点”**。

8.9 FILE REGISTERS 窗口

File Register 窗口显示选定器件的所有文件寄存器。当文件寄存器的值改变时或处理器被查询时，会更新 File Register 窗口中的数据。

> 注： 要加快使用某些硬件工具进行调试的速度，请关闭此窗口。而使用 SFR 或 Watch 窗口。

- File Registers 窗口显示
- File Registers 窗口菜单
- File Registers 窗口 FAQ

8.9.1 File Registers 窗口显示

通过单击文件寄存器窗口底部的几个按钮之一，可以改变该窗口中数据的显示方式。

- Hex
- Symbolic
- XY Data （XY 数据）（仅适用于 dsPIC 器件）

8.9.1.1 HEX

此格式将文件寄存器信息显示为十六进制数据。该窗口将具有以下列：

- Address——下一列中数据的十六进制地址。
- *Data Blocks*——十六进制数据，以 1 个字节或 2 个字节的块显示。
- ASCII——数据对应行的 ASCII 表示。

8.9.1.2 SYMBOLIC

此格式用符号显示每个文件寄存器中十六进制、十进制、二进制和字符格式的数据。该窗口将具有以下列：

- Address——数据的十六进制地址。
- Hex——十六进制数据，以 1 个字节或 2 个字节的块显示。
- Decimal （十进制）——十进制格式的数据。
- Binary （二进制）——二进制格式的数据。
- Char （字符）——字符格式的数据。
- Symbol Name （符号名称）——数据的符号名称。

8.9.1.3 XY DATA （XY 数据）（仅适用于 dsPIC 器件）

此格式将文件寄存器信息显示为十六进制数据。该窗口将具有以下列：

- Address——数据的 X 空间十六进制地址。
- Y Bus （Y 总线）——数据的 Y 空间十六进制地址 （若支持）。
- *Data Blocks*——十六进制数据，以 2 个字节的块显示。
- ASCII——数据对应行的 ASCII 表示。

欲知更多有关 dsPIC 器件的信息，请参阅 《dsPIC30F 系列参考手册》（DS70046C_CN）。

8.9.2 File Registers 窗口菜单

以下是 File Registers 窗口右键菜单中的菜单项。

- Close

  关闭此窗口。
- Full Memory Update （完全更新存储器）

  在启用了暂停或单步运行之后更新该窗口中的全部内容。

  默认情况下，此选项是启用的。如果该窗口打开，仅更新可见的数据。如果该窗口关闭，则不会更新任何数据。
- Find

  在此窗口中查找在 Find 对话框中指定的文本。
- Find Next

  查找 “查找文本”的下一个实例。

  F3 重复上一次查找。

  Shift+F3 按上一次查找的相反方向查找。
- Go To

  转到在 Go To 对话框中指定的地址。
- Import Table

  打开 Import 对话框 （参见第 **9.8** 节 **“File Management** 对话框**”**）。
- Export Table

  打开 Export As 对话框 （参见第 **9.8** 节 **“File Management** 对话框**”**）。
- Fill Registers

  用 Data 框中的值填充从 Start Address （起始地址）到 End Address （结束地址）范围内的寄存器。参见第 **9.9** 节 **“Fill Memory/Registers** 对话框**”**。
- Output to File

  将在窗口中显示的内容写入一个文本文件。使用 “另存为”对话框 （参见第 **9.8** 节 **“File Management** 对话框**”**），此外还要在该对话框的 “Output Range”中选择范围的类型，“Lines”或 “Address”，然后输入 “Start”和 “End”的值，以指定输出范围。
- Print

  打印此窗口中的内容。
- Refresh

  刷新此窗口中的数据。
- Properties

  为 SRF 和未分配的存储器选择背景颜色。还可以设置窗口的常规属性 （参见第 **9.17** 节 **“Properties** 对话框**”**）。

8.9.3 File Registers 窗口 FAQ

如何：

- 用一个值填充所有寄存器？

  在该窗口中右击并选择 “Fill Registers”以打开一个可以在其中输入填充数据的对话框。

8.10 EEPROM 窗口

EEPROM 窗口会为任何具有 EEPROM 数据存储器的单片机 （例如， PIC16F84A） 显示 EEPROM 数据。以十六进制格式显示选定器件的数据 / 操作码信息。当 EEPROM 寄存器的值改变时或处理器暂停时，会更新 EEPROM 窗口中的数据。

> 注： 为了使用编程器，需要指定 EEPROM 数据存储器的起始地址。对于大多数 PICmicro MCU 的 EEPROM 数据存储器而言，起始地址应该是 0x2100 （org H'2100'）。对于 PIC18FXXXX 器件而言，起始地址应该是 0xF00000 （org H'F00000'）。请检查选定器件的编程规范以确定正确的地址。

- EEPROM 窗口显示
- EEPROM 窗口菜单
- EEPROM 窗口 FAQ

8.10.1 EEPROM 窗口显示

该窗口在以下列中显示数据：

- Address——下一列中数据的十六进制地址。
- Data Blocks——十六进制数据，以 1 个字节或 2 个字节的块显示。
- ASCII——数据对应行的 ASCII 表示。

8.10.2 EEPROM 窗口菜单

以下是 EEPROM 窗口右键菜单中的菜单项。

- Close

  关闭此窗口。
- Find

  在此窗口中查找在 Find 对话框中指定的文本。
- Find Next

  查找 “查找文本” 的下一个实例。

  F3 重复上一次查找。

  Shift+F3 按上一次查找的相反方向查找。
- Go To

  转到在 Go To 对话框中指定的地址。
- Import Table

  打开 Import 对话框 （参见第 **9.8** 节 **“File Management 对话框”**）。
- Export Table

  打开 Export As 对话框 （参见第 **9.8** 节 **“File Management 对话框”**）。
- Fill Memory

  用 Data 框中的值填充从 Start Address 到 End Address 范围内的存储区。参见第 **9.9** 节 **“Fill Memory/Registers 对话框”**。
- Output to File

  将在窗口中显示的内容写入一个文本文件。使用 “另存为” 对话框 （参见第 **9.8** 节 **“File Management 对话框”**），此外还要在该对话框的 “Output Range” 中选择范围的类型，“Lines” 或 “Address”，然后输入 “Start” 和 “End” 的值，以指定输出范围。
- Print

  打印此窗口中的内容。

- Refresh
  刷新此窗口中的数据。
- Properties
  设置此窗口的属性。参见第 **9.17** 节 **"Properties** 对话框**"**。

8.10.3 EEPROM 窗口 FAQ

如何：

- 用值填充 EEPROM 存储器？
  在该窗口中右击并选择 "Fill Registers" 以打开一个可以在其中输入填充数据的对话框。
  如果正在使用硬件调试工具，可能还需要其他步骤，才能写片上的 EEPROM。参见所用工具的文档。

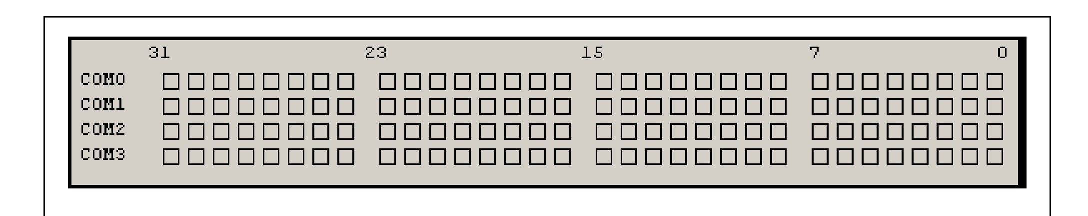
8.11 LCD PIXEL 窗口

当使能器件的 LCD 功能时， LCD Pixel 窗口将会显示 LCD 输出 （窗口的上半部分显示 LCD 输出，下半部分显示 SFR）。当禁止 LCD 功能时，所有的 LCD 像素 （方块）将以灰色作为背景色。当使能该功能时，像素将呈现白色 （未点亮或 0）或深灰 （点亮或 1）。

- LCD Pixel 窗口显示
- LCD Pixel 窗口菜单
- LCD Pixel 窗口 FAQ

8.11.1 LCD Pixel 窗口显示

对于支持 LCD 输出的器件，可以通过选择 <u>*View>LCD Pixel*</u> 来查看 LCD 输出的结果。



- LCD 显示区域
  LCD 禁止：所有的 LCD 像素将以灰色作为背景色
  LCD 使能：未点亮 （0）时， LCD 像素将呈现白色；点亮时 （1）时， LCD 像素将呈现深灰色

> **注：** 必须通过 LCD 控制寄存器中的位 （例如，对于 PIC16C924，将 LCDCON 寄存器的 bit 7 （LCDEN）置 1）才能使能 LCD 功能。请查阅具体器件的数据手册。

- Address——与 LCD 相关的 SFR 的地址。单击以改变地址的排列顺序，即由高到低或由低到高。
- SFR Name——与 LCD 功能相关的特殊功能寄存器的符号名称。
- Hex——SFR 内容的十六进制值。
- Decimal——SFR 内容的十进制值。
- Binary——SFR 内容的二进制值。
- Char——SFR 内容的 ANSI 字符值。

8.11.2 LCD Pixel 窗口菜单

以下是 LCD 显示窗口右键菜单中的菜单项。

- Close

  关闭此窗口。

- Bitfield Mouseover （鼠标移过时显示位域）

  启用 / 禁用 Bitfield Mouseover。启用该功能后，当鼠标移动到窗口中某个符号的上方时，将会弹出一条消息以显示该符号的位域。

- Find

  在此窗口中查找在 Find 对话框中指定的文本。参见第 **9.11** 节 **"Find and Replace 对话框"**。

- Find Next

  查找"查找文本"的下一个实例。

  F3 重复上一次查找。

  Shift+F3 按上一次查找的相反方向查找。

- Go To

  转到在 Go To 对话框中指定的地址。

- Fill Registers

  用 Data 框中的值填充从 Start Address 到 End Address 范围内的寄存器。参见第 **9.9** 节 **"Fill Memory/Registers 对话框"**。

- Output to File

  将在窗口中显示的内容写入一个文本文件。

- Print

  打印此窗口中的内容。

- Refresh

  刷新此窗口中的数据。

- Properties

  设置此窗口的属性。参见第 **9.17** 节 **"Properties 对话框"**。

8.11.3 LCD Pixel 窗口 FAQ

如何：

- 使能 LCD 操作？

  查阅器件的数据手册以确定设置 LCD 功能所需的 LCD 控制寄存器 （例如，对于 PIC16C924，将 LCDCON 寄存器的 bit 7 （LCDEN）置 1 来使能 LCD 功能）。可以在代码中设置这些用于 LCD 操作的寄存器；或者在窗口的下半部分找到这些 SFR 并单击 SFR 的值字段 （十六进制、十进制、二进制或字符），然后输入相应的值。

- 点亮或不点亮 LCD 像素？

  查阅器件的数据手册以确定 LCD 像素控制寄存器。可以在代码中设置这些用于 LCD 操作的寄存器；或者在窗口的下半部分找到这些 SFR 并单击 SFR 的值域 （十六进制、十进制、二进制或字符），然后输入相应的值。对应值应出现在此窗口上半部分的 LCD 输出显示中。

8.12 WATCH 窗口

Watch 窗口允许您在程序运行时监视程序中的符号。在此窗口的不同标签上至多可以设置四个不同的观察视图。

- Watch 窗口显示
- Watch 窗口菜单
- Watch 窗口 FAQ

8.12.1 Watch 窗口显示

数据显示在以下列中：

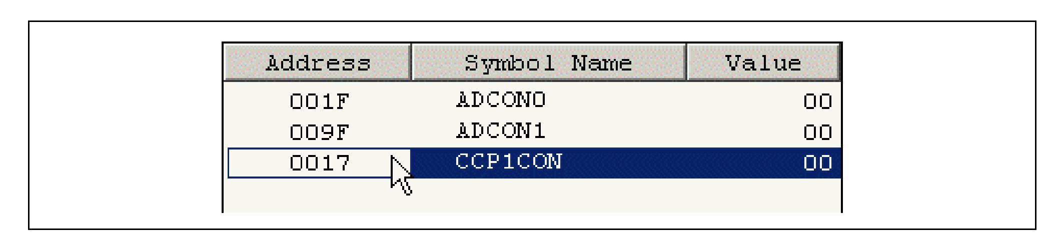
- Address——SFR 或变量的十六进制地址。

  要将绝对地址添加到 Watch 视图中，单击 Address 列的第一行并输入一个十六进制地址。
- Symbol Name——SFR 或变量的名称。

  要在 Watch 视图中添加特殊功能寄存器，请从下拉菜单中选择要添加的 SFR，然后单击 **Add SFR**。要在 Watch 视图中添加变量，请从下拉菜单中选择要添加的变量，然后单击 **Add Symbol**。
- Value——SFR 或变量的当前值。

  要更改 Watch 视图中的值，单击 Value 列中相应的行，并输入新值。然后在该列外单击以查看值是否更新。

通过右击列标题，可以在窗口中添加基数信息 （例如，十六进制、十进制、二进制或字符）。



Address	Symbol Name	Value
001F	ADCON0	00
009F	ADCON1	00
0017	CCP1CON	00

使用拖放可以重新排列 Watch 变量。还可以将 SFR 的名称或变量从 SFR、 File Register 或 Editor 窗口拖放到 Watch 窗口中。反之，也可以将它们从 Watch 窗口拖放到 Editor 窗口中。

Watch 窗口现在能对所有列进行分类。



通过单击窗口左下方的按钮，可以选择或设置四个 Watch 视图中的任何一个。

Watch 窗口——在 C 语言中的用途

如果正在用 C 开发应用程序代码，则可以使用以下功能：

- 可以使用鼠标移过时显示位域的值的功能。在 Watch 窗口中使用右键菜单可以启用 / 禁用此功能。要使用此功能， Watch 窗口必须是活动窗口 （聚焦）。
- 单击 Symbol Name 域并输入一个指针的名称，例如，`*count`。该变量必须是一个指向固有类型或结构的指针。
- 单击 Symbol Name 域并输入一个结构成员的名称，例如， `porta.ra0`。该变量必须具有 `struct.membername` 的形式。
- 单击 Symbol Name 域并输入一个结构指针成员的名称，例如， `porta->pin`。该变量必须具有 `struct->pointername` 的形式。

可以输入、复制 / 粘贴或拖 / 放符号名称。也可以从 SFR/Symbol 列表中选择符号名称，然后通过编辑符号名称来添加额外的信息。

> **注：** 对于 MPLAB C30 的 `long long int` 型 Watch 窗口将只显示 32 位，而对于 MPLAB C18 的 `short long int` 型只显示 16 位。通过 Watch 对话框改变显示。

例 1：

```
int J = 2;
int* pint = &J;
```

在 Watch 窗口中输入：`*pInt`，Value：2

例 2：

```
char[] string = "Test";
char* pstring = string;
```

在 Watch 窗口中输入：`*pstring`， Value：“Test”

例 3：

```
typedef struct
{
int str_int;
char str_ch;
} TestStruct;
TestStruct struct1;

TestStruct* pstruct1 = &struct1;
pstruct1->str_ch = 'A';
```

在 Watch 窗口中输入：`*pstruct1`， Value：结构各成员的值。

在 Watch 窗口中输入：`struct1.str_int`， Value：'A'

8.12.2 Watch 窗口菜单

以下是 Watch 窗口右键菜单中的菜单项。

- Close

  关闭此窗口。
- Find

  在此窗口中查找在 Find 对话框中指定的文本。
- Find Next

  查找 “查找文本” 的下一个实例。

  F3 重复上一次查找。

  Shift+F3 按上一次查找的相反方向查找。
- Add

  将一个 Watch 项添加到当前选定的 Watch 标签中。参见第 **9.3** 节 **“Add Watch 对话框”**。
- Delete

  将选定的 Watch 项从当前选定的 Watch 标签中删除。
- Save Watch Tab （保存观察标签）

  将当前选定的 Watch 标签中的内容保存到一个文件。
- Load Watch Tab （加载观察标签）

  将含有以前保存的 Watch 信息的文件的内容载入当前选定的标签。
- Add Watch Tab （添加观察标签）

  将一个 Watch 标签添加到 Watch 窗口。至多可以有 16 个 Watch 标签。
- Rename Watch Tab （重命名观察标签）

  重命名当前选定的 Watch 标签。
- Remove Watch Tab （删除观察标签）

  删除当前选定的 Watch 标签。 Watch 标签不能少于四个。
- Import Table

  将 Watch 数据保存到表中。参见第 **9.23** 节 **“Table Setup 对话框”**。
- Export Table

  从表装载 Watch 数据。参见第 **9.23** 节 **“Table Setup 对话框”**。
- Output to File

  将在窗口中显示的内容写入一个文本文件。
- Print

  打印此窗口中的内容。
- Refresh

  刷新此窗口中的数据。
- Properties

  打开 Watch 对话框，您可以设置字体和颜色以及 Watch 窗口的其他属性。参见第 **9.26** 节 **“Watch 对话框”**。

8.12.3 Watch 窗口 FAQ

如何：

- 添加一个观察项 （地址、 SFR 或变量）到观察窗口？
  通过单击窗口底部的相应按钮选择所需的 Watch 窗口。
  要在特定**地址**添加观察项，单击 Address 列的第一行并输入一个十六进制地址。
  要在 Watch 视图中添加**特殊功能寄存器**，请从 Add SFR 按钮旁的列表中选择相应的 SFR，然后单击该按钮。要在 Watch 视图中添加**变量**，请从 Add Symbol 按钮旁的列表中选择相应的变量，然后单击该按钮。
- 将 Watch 内容保存到文件？
  在窗口中右击以打开一个菜单。选择 Save Watch 将当前选定的 Watch 内容保存到一个文件。或者，选择 Export 将 Watch 值保存到一个文件。
- 加载一个以前保存的 Watch 窗口？
  在窗口中右击以打开一个菜单。选择 Load Watch （加载观察）将以前保存的观察内容载入当前选定的 Watch 窗口。或者，选择 Import 加载 Watch 值。
- 改变观察项的顺序？
  直接将项拖放到观察列表中希望放置的位置。
- 观察寄存器的单个位？
  参见第 **9.26 节 “Watch 对话框”**。
- 观察 C 指针和结构成员？
  参见 Watch 窗口——在 C 语言中的用途。

8.13 SPECIAL FUNCTION REGISTERS 窗口

Special Function Registers （特殊功能寄存器）窗口为选定处理器显示特殊功能寄存器 （SFR）中的内容。

> **注：** 如果器件上实际并不存在某个数据存储寄存器，该寄存器可能就不会出现在 SFR 列表中。有些工具 （如软件模拟器），可能允许您查看在实际器件上并不存在的寄存器，如预分频器。

此窗口提供的格式在查看 SFR 时比使用普通文件寄存器窗口查看更有用，因为它包含了每个 SFR 名称并提供了几种编号格式。每当发生间断时，均会更新特殊功能寄存器的内容。

> **注：** 如果选择了 “Freeze Peripherals On Halt”（暂停时冻结外设），单步运行时将不会更新 SFR 或观察窗口中的 I/O 端口位。引脚值将被更改，但是读取新值的请求会被冻结阻止并且数据将不能被更新，直到程序运行到下一步或执行运行命令为止。

- SFR 窗口显示
- SFR 窗口菜单
- SFR 窗口 FAQ

8.13.1 SFR 窗口显示

在以下列中显示数据。

- Address——SFR 的十六进制地址。
- SFR Name——SFR 的符号名称。
- Hex——十六进制值，以 1 个字节的块显示。
- Decimal——十进制格式的值。
- Binary——二进制格式的值。
- Char——字符格式的值。

8.13.2 SFR 窗口菜单

以下是 Special Function Registers 窗口右键菜单中的菜单项。

- Close

  关闭此窗口。
- Bitfield Mouseover

  启用 / 禁用鼠标移过时显示位域值的功能。要使用此功能， SFR 窗口必须是活动窗口 （聚焦）。
- Find

  在此窗口中查找在 Find 对话框中指定的文本。
- Find Next

  查找 “查找文本”的下一个实例。

  F3 重复上一次查找。

  Shift+F3 按上一次查找的相反方向查找。
- Go To

  转到在 Go To 对话框中指定的地址。
- Fill Registers

  用 Data 框中的值填充从 Start Address 到 End Address 范围内的寄存器。参见第 **9.9** 节 **“Fill Memory/Registers 对话框”**。

注：	并非所有的 SFR 都是可写的，某些 SFR 是镜像的。

- Output to File

  将在窗口中显示的内容写入一个文本文件。使用 “另存为”对话框 （参见第 **9.8** 节 **“File Management 对话框”**），以及 “Output Range”和 Tab Delimited （制表符定界）复选框。选择范围的类型，“Lines”或 “Address”，然后输入 “Start”和 “End”值，以指定输出的范围。
- Print

  打印此窗口中的内容。
- Refresh

  刷新此窗口中的数据。
- Properties

  设置此窗口的属性。参见第 **9.17** 节 **“Properties 对话框”**。

8.13.3 SFR 窗口 FAQ

如何：

- 用一个值填充所有寄存器？

  在该窗口中右击并选择 “Fill Registers”以打开一个可以在其中输入填充数据的对话框。

8.14 TRACE MEMORY 窗口

View 菜单下的 Trace Memory 窗口可以帮助您监视大多数的处理器操作。至多可以显示 32767 个指令周期。

- Trace 窗口显示
- Trace 窗口菜单
- Trace 窗口 FAQ

8.14.1 Trace 窗口显示

Trace Memory 窗口包含下列有关每个执行周期的信息：

- 行号 （**Line**）——相对于触发点或暂停点的周期位置。
- 地址 （**Addr**）——从程序存储器中取出的指令的地址。
- 操作码 （**Op**）——取出的指令。
- 标号 （**Label**）——与程序存储器地址关联的标号 （如果有）。
- 指令 （**Instruction**）——反汇编指令。
- 源数据地址 （**SA**）——源数据的地址或符号 （如果适用）。
- 源数据值 （**SD**）——源数据的值 （如果适用）。
- 目标数据地址 （**DA**）——目标数据的地址或符号 （如果适用）。
- 目标数据值 （**DD**）——目标数据的值 （如果适用）。
- 时间戳 （**Cycles/Time**）——以周期数或秒数表示的时间标记的值。
- 外部输入 （**Probe #**）——外部输入的值 （如果有）。

一般情况下，会将触发周期编号为周期 0。所有其他周期将以此周期为基准编号。发生在触发点之前的周期将有负的周期号，发生在触发点之后的周期将有正的周期号。

Trace memory 窗口可始终保持打开，并可移动或调整其大小。还可 “就地”编辑该窗口中的数据。

8.14.2 Trace 窗口菜单

以下是 Trace Memory 窗口右键菜单中的菜单项。

- Close

  关闭此窗口。
- Find

  打开 Find 对话框。在 Find What 字段中，输入想要查找的一串文本，或从下拉列表中选择文本。在打开 Find 对话框之前，也可以在编辑窗口中选择文本或用光标覆盖想要搜索的字。

  在 Find （查找）对话框中，可选择任何可用的选项以及希望搜索的方向。 Up （向上）从插入点反向搜索， Down （向下）从插入点正向搜索。
- Find Next

  查找 “查找文本”的下一个实例。

  F3 重复上一次查找。

  Shift+F3 按上一次查找的相反方向查找。

- Go To
  跳到对话框中指定的行。行号显示在该窗口的第一列中。
- Go To Source Line （转到源代码行）
  转到与选定跟踪代码行相对应的源代码行。
- Jump to Trigger （跳至触发）
  跳至触发的位置。
- Jump to Top （跳至顶端）
  跳至该窗口的顶端。
- Jump to Bottom （跳至底部）
  跳至该窗口的底部。
- Reload （重载）
  用跟踪缓冲器的内容重载跟踪存储器窗口。
- Reset Time Stamp （复位时间标记）
  可在处理器复位时、运行时有条件地复位时间标记或手动复位时间标记。或者，通过选择 Reset Now （立即复位）强制立即复位。
- Display Time （显示时间）
  以经过的秒数或工程学格式，以周期数显示时间标记。
- Symbolic Disassembly （符号反汇编）
  在列表中显示名称以取代 SFR 和变量的数值地址。
- Output to File
  将跟踪存储器窗口的内容导出到一个文件。使用 “另存为”对话框 （参见**第 9.8 节 “File Management 对话框”**），以及周期和制表符信息。输入 “Start”和 “End”周期以指定要写入文件的周期范围。还要指定该文本是否采用制表符定界。
- Print
  打印跟踪存储器窗口中的内容。
- Refresh
  刷新此窗口中的可视内容。
- Properties
  设置此窗口的属性。参见**第 9.17 节 “Properties 对话框”**。

8.14.3 Trace 窗口 FAQ

如何：

- 设置简单的跟踪？
  在含有应用程序代码的文件 （编辑器）窗口中，从右键菜单中选择 “Add Filter-in Trace”（添加选入跟踪）或 “Add Filter-out Trace”（添加排除跟踪）。欲知更多有关筛选跟踪的信息，参见**第 11.5 节 “使用调试功能”**。
- 设置较为复杂的跟踪？
  许多调试工具允许您设置条件来定义跟踪。查阅该工具的文档以确定如何设置这种类型的跟踪。

8.15 CONFIGURATION BITS 窗口

Configuration Bits 窗口显示有关配置位设置的信息。 MPLAB IDE v6.xx 现在能识别在代码中以及在 Configuration Bits 窗口中设置的配置位。任何未在代码或此窗口中设置的配置位，其值默认为 1。

加载程序时 （即编译、导入或打开项目），在代码中做出的配置位设置将反映在该窗口中。在该窗口中做出的配置位设置也将立即反映出来。在调试器执行代码期间， MPLAB IDE 会使用该窗口的当前值。

如果在 *Configure>Settings*， **Program Loading** 选项卡中选中了 “Clear program memory upon loading a program”（加载程序时清除程序存储器），则配置位将被清零，然后在加载程序时加载已经过更改的代码。如果在代码中未指定任何配置位设置，则会将这些位直接清零。

关闭一个工作区时， MPLAB IDE 将最新的配置位信息保存在工作区文件中。打开工作区时， MPLAB IDE 会清除存储器，加载用于设置配置位的代码，然后加载上一次保存在工作区中的 Configuration Bits 窗口的信息。

- Configuration Bits 窗口显示
- Configuration Bits 窗口菜单
- Configuration Bits 窗口 FAQ

8.15.1 Configuration Bits 窗口显示

选择 *Configure>Configuration Bits* 打开 Configuration Bits 窗口。使用此窗口为选定器件设置配置位的值。

注：	这些值是在编译源文件时获得的，它们用于控制调试工具的操作。

- Address——配置字 / 字节的地址。
- Value （值）——配置字 / 字节的当前值。
- Category （类别）——相应配置字 / 字节中配置位的名称。
- Setting （设置）——配置位的当前设置。使用下拉列表可更改设置。配置字 / 字节的值也将相应地改变。

8.15.2 Configuration Bits 窗口菜单

以下是 Configuration Bits 窗口右键菜单中的菜单项。单击一项以展开或折叠描述。

- Close

  关闭此窗口。
- Reset to Defaults （复位为默认设置）

  还原该窗口的默认设置。这将符合实际器件上电时配置位的状态。
- Refresh

  刷新此窗口中的数据。
- Properties

  设置窗口的属性。参见第 **9.17** 节 **“Properties** 对话框**”**。

8.15.3 Configuration Bits 窗口 FAQ

如何：

- 在加载程序时清零配置位？

  选择 *Configure>Settings* 并单击 **Program Loading** 选项卡。选中 Clear configuration bits upon program loading （加载程序时清零配置位）复选框。
- 使用 Configuration Bits 窗口开发应用程序？

  如果只使用 Configuration Bits 窗口来开发应用程序：
  - 不要在代码中设置任何配置位，即不使用初始化数据。
  - 应清除 *Configure>Settings*，**Program Loading** 选项卡上的“Clear program memory upon loading a program”复选框。

  完成开发之后，还必须将 Configuration Bits 窗口设置复制到初始化数据中。
- 使用在代码中设置的配置位开发应用程序？

  如果只使用代码中的配置位 （即，初始化数据）来开发应用程序：
  - 不要更改 Configuration Bits 窗口中的任何配置位设置。
  - 应选中 *Configure>Settings*，**Program Loading** 选项卡上的“Clear program memory upon loading a program”复选框。

8.16 文件 （编辑器）窗口

文件窗口包含应用程序的源代码。您可以在此窗口中打开现有的源代码文本文件（*File>Open*），或打开一个空白的窗口并输入您的代码 （*File>New*）。一旦将文件包含在项目中，如果在保存项目时让该文件处于打开状态，那么每次打开该项目时（*Project>Open*）都会打开该文件。

使用 MPLAB 编辑器可编辑文件窗口中的文本。更多有关该编辑器操作的信息，请参见相关文档。

- 文件窗口显示
- 文件窗口菜单
- 文件窗口 FAQ

8.16.1 文件窗口显示

通过单击并拖动滚动箭头前的矩形 “小块”（nub）可以拆分文件窗口 （参见下图）。拆分窗口使您能同时查看长代码的不同部分。



小块（Nub）

此窗口的右键菜单包含一些文本编辑和调试选项。在 Edit 菜单中可以找到更多的文本编辑选项 （更多信息参见 MPLAB 编辑器帮助）。在 Debugger 菜单上可以找到更多调试选项 （选定调试工具之后）。

代码执行信息可能会以符号的形式显示在该窗口左侧的灰色区域中。

若将鼠标或光标移动到一个变量的上方时，将弹出变量的地址。

8.16.2 文件窗口菜单

以下是 File 窗口右键菜单中的菜单项。

- Remove All Filter Traces （删除所有筛选跟踪）
  删除代码文本上的所有筛选跟踪标记。参见第 **11.5** 节 **"使用调试功能"**。
- Add Filter-in Trace （添加选入跟踪）
  在选定的代码文本上添加选入跟踪标记。参见第 **11.5** 节 **"使用调试功能"**。
- Add Filter-out Trace （添加排除跟踪）
  在选定的代码文本上添加排除跟踪标记。参见第 **11.5** 节 **"使用调试功能"**。
- Remove Filter Trace （删除筛选跟踪）
  在选定的代码文本上删除筛选跟踪标记。参见第 **11.5** 节 **"使用调试功能"**。
- Close
  关闭活动窗口。
- Set/Remove Breakpoint
  在当前选定的行上设置或删除断点。
- Enable/Disable Breakpoint
  在当前选定的行上启用或禁用断点。
- Breakpoints
  禁用、启用或删除所有断点。
- Run to Cursor
  将程序运行到当前的光标位置。在以前的版本中称为 Run to Here （运行到此）。
- Set PC at Cursor
  将程序计数器 （PC）设置到光标位置。
- Cut
  删除在当前窗口中选定的文本并将它放在剪贴板上。在此操作之后，可将被删除的文本粘贴到另一个 MPLAB 编辑器窗口、同一个 MPLAB 编辑器窗口中的另一个位置或另一个 Windows 应用程序中。
- Copy
  将当前窗口中选定的文本复制到剪贴板。在此操作之后，可将所复制的文本粘贴到另一个 MPLAB 编辑器窗口、同一个 MPLAB 编辑器窗口的另一个位置或另一个 Windows 应用程序中。
- Paste
  将剪贴板的内容粘贴到当前窗口的插入点处。仅当剪贴板包含文本格式的数据时才可以执行此操作。 MPLAB 编辑器不支持粘贴位图或其他的剪贴板格式。
- Delete
  删除选定的文本。
- Add to Project
  将文件插入到当前项目中。 MPLAB IDE 将根据文件的类型将它们归类到项目窗口树下正确的类型中。

- Advanced （高级）

  设置高级的文本功能。将选定文本变为全大写或全小写、注释或非注释、缩进或伸出，如果选定文本为花括号、方括号或圆括号时，还可查找与之匹配的另一个括号。
- Bookmark

  管理书签。切换 （启用 / 禁用）书签，移动到下一个或上一个书签或者清除所有的书签。更多有关书签的信息，请参见 MPLAB 编辑器在线帮助。
- Text Mode （文本模式）

  根据开发模式 （即器件架构和编程语言）定制文本显示。
- Properties

  设置 Editor Options 中的显示或功能选项。参见第 **11.2.1** 节 **"Editor Options** 对话框**"**。

8.16.3 文件窗口 FAQ

如何：

- 根据代码的类型设置代码的颜色？

  MPLAB IDE 将自动为您的代码设置颜色。要更改此设置，请在窗口中右击，然后选择 Text Mode，再选择所需的类型，类型可以是器件特定的汇编、 C、 Basic 或 SCL （软件模拟器控制语言）。
- 设置代码的颜色？

  在窗口中右击并选择 Properties。在 Editor Options 对话框中，单击 **Text** （文本）选项卡，然后单击 **Choose Colors** （选择颜色）。



第 9 章 MPLAB IDE 对话框

9.1 简介

MPLAB IDE 对话框的操作和普通的 Windows 应用程序基本相同。下面列出了 MPLAB IDE 中的基本对话框。根据已经安装的工具，可能还可使用其他对话框。

本章论及的对话框按字母顺序排列如下：

- About MPLAB IDE 对话框
- Add Watch 对话框
- Breakpoints 对话框
- Build Options 对话框
- Export Hex File 对话框
- External Memory Setting 对话框
- File Management 对话框 （打开、另存为和添加文件）
- Fill Memory/Registers 对话框
- Find In Project Files 对话框
- Find and Replace 对话框
- Help Topics 对话框
- Import 对话框
- New Project 对话框
- Project-Display Preferences 对话框
- Project Wizard 对话框
- Properties 对话框
- Save Project As 对话框
- Select Device 对话框
- Select Language Toolsuite 对话框
- Set Language Tool Location 对话框
- Settings 对话框
- Table Setup 对话框
- User ID Memory 对话框
- Version-Control 对话框
- Watch 对话框

9.2 ABOUT MPLAB IDE 对话框

选择 *Help>About MPLAB IDE* 以打开 About MPLAB IDE 对话框。这是一个信息对话框。

单击 **OK** 关闭对话框。

商标信息

第一部分包含 Microchip 产品的商标信息。此外还讨论了其他产品的商标信息。

组件和插件信息

该列表框包含以下信息：

- 名称——组件 / 插件的名称
- 版本——组件 / 插件的版本号
- MPLAB 认证——组件 / 插件是否通过了 MPLAB 认证。欲知更多有关认证的信息，请联络 joe.drzewiecki@microchip.com。

单击列表中的项，就会在 "Module path"（模块路径）中显示该模块的路径。

9.3 ADD WATCH 对话框

右击 Watch 窗口并选择 **Add** 来打开 Add Watch 对话框。使用此对话框将观察项添加到当前选定的 Watch 选项卡中。在完成添加项后，单击 **Close**。

相关的窗口 / 对话框是：

- 第 **8.12** 节 **"Watch** 窗口**"**
- 第 **9.23** 节 **"Table Setup** 对话框**"**
- 第 **9.26** 节 **"Watch** 对话框**"**

对话框	说明
Add an SFR	从下拉菜单中选择一个特殊功能寄存器 （SFR）的名称。然后单击 **Add SFR**。
Add a Symbol	从下拉菜单中选择一个变量名称。然后单击 **Add Symbol**（添加变量）。
Add an Absolute Address or Address Range （添加绝对地址或一个地址范围）	首先选择一种存储器类型。可能的地址范围就会被显示在 Range 中。然后，选择输入地址的格式。最后，输入 Start Address 和 End Address 或者 Word Count （字数）。单击 **Add Address** （添加地址）。

9.4 BREAKPOINTS 对话框

选择 *Debugger>Breakpoints* 以打开 Breakpoints 对话框。必须首先选择调试工具，此选项才会出现在 Debugger 菜单中。

> 注： 还有其他方法可以设置断点。参见**第 3.18 节 "使用断点"**。

对话框	说明
Break At:（在此处设置断点：）	输入断点所在的源代码行号或地址。单击 <Enter> （回车键）将断点放置在下面的列表中。
Program Memory Breakpoints （程序存储器断点）	设置在程序存储器中的断点的列表。可以选择这些断点，然后通过右边的按钮对这些断点进行操作。
Active Breakpoint Limit （有效断点限制）	对于选定调试工具来说，可以设置的断点总数。
Available Breakpoints （可用断点）	在减去代码断点和高级断点 （若适用）之后剩下的断点数量。 **注意：**"Run to Cursor"和 "Step Over"命令使用断点运行和停止。若没有可用断点，这些命令将单步运行并比较，直到到达停止的位置，这种情况会使这些命令执行的更加缓慢。

输入断点

在 "Break At"框中输入断点的位置。

- 输入某个代码行号，例如， 27，还必须输入源文件的路径和名称才能真正的设置断点，例如， `c:\project1\program1.asm, 27`。

- 输入某个代码行的地址 （十六进制），例如， `002A`。

您将看到输入的断点出现在 "Program Memory Breakpoints"框中。在输入了断点之后，单击回车键将看到该断点已被输入到 "Program Memory Breakpoints"框中，并且在断点的前面会出现一个复选框，这样即可启用 / 禁止此断点。

删除断点

- 在 "Program Memory Breakpoints"框中单击断点以便标识断点。然后单击 **Remove** 按钮。

删除所有断点

- 单击 **Remove All** （删除所有断点）按钮将所有断点从 "Program Memory Breakpoints"框中删除。

启用或禁用断点

- 单击断点前面的复选框可交替地启用 / 禁用断点。

启用或禁用所有断点

- 单击 **Enable All** （启用所有断点）或 **Disable All** （禁用所有断点）可分别启用或禁用 "Program Memory Breakpoints"框中列出的所有断点。

修复无法解析的断点

如果已经在一行高级语言代码 （如 C 代码）上设置了断点，但是无法解析该代码 （找不到对应的汇编代码），您会收到一条警告消息。并且会在断点对话框中，无法解析的断点旁边出现一个黄色的问号。

优化过的代码通常会导致这种问题。重新编译应用程序，但并不优化代码，然后再次尝试解析断点。

另一种解决方案就是直接将断点移到另一个可被解析的代码行上。

保存 / 取消更改

单击 **OK** 保存并实现改变，然后关闭该对话框。单击 **Cancel** 也可关闭该对话框，但并不保存或实现改变。

9.5 BUILD OPTIONS 对话框

选择 *Project>Build Options>Project* 以打开 Build Options 对话框。必须首先打开一个项目，才可从 Project 菜单中选择该选项。

9.5.1 General 选项卡

输入或查找目录，也可搜索 MPLAB IDE 在编译项目的时候将要使用的路径。目录是文件的一个特定路径，而搜索路径可能包括几个路径，它们由分号 （；）分开。其他相关的路径 （如 `..\tmp`）与源文件的位置有关。

> **注：** 仅可输入 MPLAB IDE 将要使用的路径，例如，若选用了 MPASM 汇编器将不会使用 Assembler Include Path （汇编器头文件路径）或 Include Path （头文件路径）信息。请查阅语言工具的文档以确定它是否使用此信息。

对话框	说明
输出目录	包含仅由完全编译，尤其是由链接产生的文件的路径。也就是 COD、 COF 和 hex 文件以及列表和映射文件的路径 。这些文件由 *Project>Clean* 清除。
中间目录	编译项目时产生的中间文件的路径。如需删除这些文件，请使用 *Project>Clean*。
汇编器头文件路径	汇编器头文件 （`.inc`）的路径。
头文件路径	头文件 （`.h`）的路径。
库路径	库文件 （`.lib`， `.arc`）的路径。
链接描述文件路径	链接描述文件 （`.lkr`， `.gld`）的目录的路径。
套用默认设置	用默认设置替换当前选择的目录和路径。默认设置可在 *Project>Set Language Tool Locations* 中设置。

9.5.2 Language Tool 选项卡

设置选定工具包中的语言工具。欲知更多信息请参阅每种工具的在线帮助。

9.6 EXPORT HEX FILE 对话框

通过 *File>Export* 可以使用 Export Hex File 对话框。

使用 Export 可将整个程序存储器、程序存储器的一部分、EEPROM 存储器、配置位或用户 ID 保存到一个 `.hex` 文件。可采用两种格式。在大多数情况下应使用 INHX32 格式。 INHX8S 格式将存储区保存为两个文件：一个用于保存偶数字节，另一个用于保存奇数字节。这对于在成对的外部 8 位宽的存储器芯片上存储程序存储器内容很有用处。

如需导出十六进制文件：

1. 请单击 **File Format** （文件格式）选项卡。选择 Hex 文件的格式。
2. 单击 **Memory Areas** （存储区）选项卡。选择要导出的存储区，并且在允许指定范围的情况下，输入存储区的范围。
3. 单击 **OK**。
4. 在 “另存为” 对话框中，输入文件的名称及其位置并单击**保存**。

9.7 EXTERNAL MEMORY SETTING 对话框

某些 PICmicro MCU 和 dsPIC DSC 器件通过使用片外 （也称为外部）存储器支持扩展的程序存储器。可选用这些器件的某些模式，比如扩展的单片机 （部分片上和部分片外程序存储器）和微处理器 （全部是片外程序存储器）。

要使得调试工具能够使用外部存储器，必须选择支持此功能的器件（*Configure>Select Device*）。然后选择 *Configure>External Memory*。这样将打开 External Memory Setting 对话框。

- **Use External Memory** （使用外部存储器）——启用 / 禁止使用外部存储器
- **Address Range**——总是将 **Start** 地址设置为内部存储器的末尾地址。然后输入外部存储器的 **End** 地址

一旦启用了外部程序存储器，它就会出现在 Program Memory 窗口中。

单击 **OK** 保存更改，并关闭该对话框。单击 **Cancel** 关闭该对话框，但不保存更改。

9.8 FILE MANAGEMENT 对话框

File management 对话框允许管理源文件和项目文件。 MPLAB IDE 具有以下类型的文件管理对话框：

输入

- **打开**对话框——打开现有的文件、项目或工作区或者导入现有的调试文件。
- **Add Files to Project** 对话框——将源文件插入到项目中。
- **Import** 对话框——将表列数据从文件导入到存储器窗口。

输出

- **另存为对话框**——以不同的名称或在不同的文件夹中保存文件、项目或工作区。
- **Export As** 对话框——将表列数据从存储器窗口导出到一个文件。

9.8.1 文件管理

控制	对话框	说明
查找文件	输入	从下拉列表中选择包含文件 / 项目的文件夹。默认为上一次查看的目录。
保存文件	输出	
文件的菜单	全部	在选定的文件夹中查找一系列的文件 / 项目。可在 “文件类型” / “保存类型”列表框中选择要显示的文件的类型。单击选中显示出的某个文件或在 “文件名”文本框中输入文件名。 此文本框具有以下编辑功能：删除文件、重命名文件和拖放复制文件。
文件名	全部	输入文件 / 项目的名称或者从文件文本框中选择一个文件。
文件类型	输入	从文件类型列表中选择希望在文件文本框中查看的文件的类型。
保存类型	输出	
起始地址	Import/Export As	指定从哪个地址开始导入 / 导出数据。
结束地址， 单列输出	Export As	指定从哪个地址停止导出数据。选择复选框 “Single Column Output”（单列输出）以进行单列输出或者不选择该复选框以进行多列输出。

9.8.2 Navigation 按钮

- 导航到访问过的上一个文件夹
- 导航到上一级目录
- 创建新文件夹
- 使用 Thumbnails、 Tiles、 Icons、 List 或 Details 菜单以多种形式查看文件。

9.9 FILL MEMORY/REGISTERS 对话框

如需用一个值填充存储器，右击以下一个窗口并选择 “Fill Memory”。

- **第 8.8 节 “Program Memory 窗口”**
- **第 8.10 节 “EEPROM 窗口”**

如需用一个值填充寄存器，右击以下一个窗口并选择 “Fill Registers”。

- **第 8.9 节 “File Registers 窗口”**
- **第 8.13 节 “Special Function Registers 窗口”**

输入 “Start Address”和 “End Address”作为填充范围。若要保存此范围使之始终出现在此对话框中，请选择 “Retain Address Range”（保留地址范围）。

在 “Data”框中输入一个要被填充到指定范围内的每个地址中的值，或者选中 “Randomize Data”（随机数据）复选框。若选中 “Sequence Start”（顺序开始）则会以有序的数据填充存储器。

选择 “Data Radix”（数据基数），可以选择 Hexadecimal，也可以选择 Decimal。

单击 **Write** （写）以填充存储器或寄存器范围。单击 **Close** 放弃。

9.10 FIND IN PROJECT FILES 对话框

选择 *Project>Find In Project Files* 打开一个对话框，该对话框允许在项目的多个文件中搜索一串文本。搜索结果显示在 Output 窗口中。

9.11 FIND AND REPLACE 对话框

使用 Find 对话框 （*Edit>Find*）查找文件 （编辑器）窗口中的一串文本。使用 Replace 对话框 （*Edit>Replace*）用另一串文本替换文件 （编辑器）窗口中的某一串文本。

项	对话框	说明
Find what：	Find、Replace	输入想要查找的一串文本，或者从下拉列表中选择一文本。也可以在文件窗口中选择文本或者在打开该对话框之前将指针放在想要搜索的字上。
Replace with： （替换为）	Replace	输入一串想用来替换找到的文本的文本，或者从下拉列表中选择一文本。
Match whole word only （整体匹配）	Find、Replace	文件窗口中的文本必须与 “Find what”中的文本整体匹配，而不是仅与一部分匹配。
Match case （匹配大小写）	Find、Replace	文件窗口中的文本必须与 “Find what”中的文本大小写匹配。例如，“prog1”和 “Prog1”就不匹配。
Direction （方向）	Find、Replace	Up 从指针的位置反向搜索； Down 正向搜索。
Find Next	Find、Replace	突出显示下一个在文中出现的 “Find What”中的文本。
Replace	Replace	用 “Replace with”中的文本替换选定的文本。
Replace All （替换所有）	Replace	用 “Replace with”中的文本替换文中所有出现在 “Find What”中的文本。

9.12 HELP TOPICS 对话框

选择 *Help>Topics* 打开 MPLAB Help Topics 对话框。使用此对话框选择要显示的帮助文件。帮助文件是根据它们所描述的开发工具的类型来分类的，即分为 System、Language Tools、 Debuggers、Programmers 和 （其他） Tools。您也许只会选择打开一个帮助文件。

双击要选择的文件，或者单击该文件然后单击 **OK** 以打开相应的帮助文件。单击 **Cancel** 关闭该对话框而不打开任何帮助文件。

9.13 IMPORT 对话框

通过 *File>Import* 可以使用 Import 对话框。它基本上是一个 “打开”对话框 （参见**第 9.8 节 “File Management 对话框”**）。

使用 Import 对话框可以将 `.hex` 文件载入存储器。 hex 文件可能包含程序存储器、数据 EEPROM、配置位和 ID 单元的数据。若要使用先前汇编或编译的项目来编程一个器件，导入操作将非常有用。如果相关的调试文件和 `.hex` 文件在同一个目录中，执行该操作还会将符号和行号信息载入当前的项目。

也可导入以下某个调试文件：`.cof` 或 `.cod` 文件。这两个文件包含来自相关 `.hex` 文件的存储器信息和项目上一次编译时的符号信息。导入完成时不会有任何通知，但是 IDE 窗口中的信息会发生改变。

9.14 NEW PROJECT 对话框

输入新项目的名称和路径 （*Project>New*）。

- Name——输入新项目的名称。
- Directory——输入新项目的目录。可使用以下一种方法来输入新项目的名称和路径：
  - 输入现有目录的路径或者输入新目录的路径，输入新路径时，单击 **OK** 会提示创建目录。
  - 单击 Browse 找到现有的目录或想要放置新目录的上一级目录。在 Folder 对话框的 Browse 中单击 **OK**。如果正在创建新目录，请指定路径然后单击 **OK**。若目录不存在会提示您创建目录。

9.15 PROJECT-DISPLAY PREFERENCES 对话框

右击项目窗口的空白区域从弹出菜单中选择 Preferences，以显示 Project-Display Preferences 对话框。使用此对话框可设置项目窗口的首选项。

对话框	说明
Display Project Nodes As （将项目节点显示为）	选择想要在项目树上显示项目文件类型的方式，可仅显示文件名，或显示文件名及其路径。
Display File Nodes As （将文件节点显示为）	选择想要在项目树上显示项目文件的方式，可仅显示文件名、文件名及其路径，或者以文件内部指定的方式显示项目文件。
Refresh Version-Controlled Files （刷新版本控制文件）	如果将项目设置为使用版本控制的文件（*Project>Select Version-Control System*），若想要在更新项目窗口时反映出版本控制状态，请选择此项。刷新意味着在更新每个文件时，均会询问版本控制系统 "谁检出了此文件？"。这可能会很慢，尤其在项目中有很多文件的情况下，更是如此。如果禁止 "Refresh on application focus"（刷新当前应用程序）和 "Auto refresh"（自动刷新），仍可在执行了版本控制操作后进行刷新，在项目窗口右键菜单上也有相应的 "Refresh" 命令。

9.16 PROJECT WIZARD 对话框

选择 *Project>Project Wizard* 以启动一系列对话框来设置您的项目。欲知更多信息，请参阅**第 4.2 节 "使用项目向导"**。

9.17 PROPERTIES 对话框

在某个窗口中右击并从弹出菜单中选择 Properties 会显示出一个对话框，可以在此设置窗口属性，如字体类型和颜色、背景色和列设置。根据窗口的不同，此对话框中的选项卡也会有所不同。下面将描述一般的选项卡。

9.17.1 Column Settings 选项卡

窗口中使用的列标题将被列在一个列表框中。

要显示 / 隐藏列

- 可选中 / 清除某个列标题旁边的复选框以显示 / 隐藏该列。
- 可以单击某列以选中它。然后使用 **Show** （显示）或 **Hide** （隐藏）按钮以显示或隐藏该列。

要调整列的顺序

- 可以单击某列以选中它。然后使用 **Move Up** （上移）或 **Move Down** （下移）按钮以上移或下移该列。

要查看列宽

- 可以单击某列以选中它。该列的宽度将被显示在 "Selected column width in pixels"（所选列的宽度 （以像素为单位））中。

要更改列宽

- 激活该窗口。
- 将光标移到两列之间的线上，直到它变为调整大小光标为止。
- 单击并拖动该线条以使列变宽或变窄。

要恢复默认设置

- 可单击 **Defaults**。

9.17.2 General 选项卡

为所有调试窗口设置字体并更改颜色，即在此处进行的设置会全局决定所有调试（View 菜单）窗口的设置。

字体

单击 **Select Font** （选择字体）以打开标准的 Windows “字体”对话框，可在此为选定字体 （例如 “西方”）设置字体类型、字形和字体大小。要在 “字体”对话框中仅选择不按比例调整的字体，可以选择 “Show Fixed Pitch Only”（仅显示固定间距的字体）。

颜色

单击 **Change Color** （改变颜色）以打开标准的 Windows “颜色”对话框，可以在此设置 “改变颜色”。“ 改变颜色”是在改变发生后被改变的文本所呈现的颜色，例如，当 `movwf PORTB` 将 PORTB 的值从 00 更改为 55 时， 55 就会用“改变颜色”显示出来。

9.18 SAVE PROJECT AS 对话框

选择 *Project>Save Project As* 以打开 Save Project As 对话框。

当项目被保存到一个新目录中时，项目目录中的所有文件和它的子目录都将和新的项目以及工作区文件一起被复制到新目录中。在项目树以外的任何文件都会被认为是共享文件而不被复制。项目窗口的上下文菜单将发生改变以反映项目的新位置。

为了让项目（及其相关的工作区）便于移动， MPLAB IDE 采用相对路径保存在项目目录及其子目录中的文件，而采用完整路径存储项目目录之外的文件。

Save Project As 对话框在功能上与标准 “另存为”对话框 （参见第 **9.8** 节 **“File Management 对话框”**）相同。

9.19 SELECT DEVICE 对话框

要选择用于开发的器件，请选择 Configure>Select Device 以打开 Select Device 对话框。

Device 下将列出此版本的 MPLAB IDE 当前所支持的所有器件。通过从列表中选择或者输入器件名称可选择器件。

一旦从此列表中选择了器件， Microchip Programmer Tool Support （Microchip 编程器工具支持）部分和 Microchip Debugger Tool Support （Microchip 调试器工具支持）部分均会反映对 Microchip Technology 器件的工具支持：

- 绿色按钮——支持
- 黄色按钮——测试阶段
- 红色按钮——目前不支持

注：	鼠标悬停在一个按钮上即会显示出与之相关的支持。

单击 **OK** 保存更改，并关闭该对话框。单击 **Cancel** 关闭该对话框，但不保存更改。

要选择编程器，请从 Programmer>Select Programmer 的列表中选择。

要选择调试工具，请从 Debugger>Select Tool 的列表中选择。

9.20 SELECT LANGUAGE TOOLSUITE 对话框

选择 Project>Select Language Toolsuite 以打开 Select Language Toolsuite 对话框。必须首先打开一个项目，才可从 Project 菜单中选择该选项。

使用此对话框选择将要在项目中使用的语言工具包。查看语言工具的文档以确保它们支持将要使用的器件。

- **Active Toolsuite**——选择将要使用的工具包。
- **Toolsuite Contents**——查看与上面选定的工具包相关的语言工具。如果没有所需的工具，请选择其他工具包。单击一个语言工具以查看其位置。
- **Location**——更改路径或文件、输入新的路径或文件信息或者单击 **Browse** 按钮以找到在上面的列表框中选定的语言工具的可执行文件。

工具旁的红 “X”表示该工具未安装或者 MPLAB IDE 无法获取该工具的路径。欲知更多信息，请参阅**第 9.21 节 “Set Language Tool Location 对话框”**。

9.21 SET LANGUAGE TOOL LOCATION 对话框

选择 Project>Set Language Tool Locations 以打开 Set Language Tool Location 对话框。

使用此对话框设置工具包中各个语言工具的可执行文件的路径。

- **Registered Tools** （已注册的工具）——查找将要使用的工具包 （例如 Microchip MPASM 工具包）。单击 “＋”号将其展开。
  - 单击 “Executables”（可执行文件）旁边的 “＋”号将其展开。从中单击一个工具以在 “Location”中查看当前为它指定的路径。
  - 单击 “Default Search Paths and Directories”（默认搜索路径和目录）旁边的 “＋”号以将其展开。从中单击一个路径以在 “Location”中查看当前指定的路径。
  - **Location**——更改路径或文件、输入新的路径或文件信息或者单击 **Browse** 按钮查找路径或文件。

9.22 SETTINGS 对话框

选择 *Configure>Settings* 以打开 Settings 对话框并设置常规的 MPLAB IDE 功能。单击 **OK** 保存更改，并关闭该对话框。单击 **Cancel** 关闭该对话框，但不保存更改。

- Workspace 选项卡——设置工作区选项
- Debugger 选项卡——设置调试选项
- Program Loading 选项卡——设置在程序装载时清零 / 不清零存储器
- Hot Keys 选项卡——为各种 IDE 操作设置热键
- Projects 选项卡——设置项目选项

9.22.1 Workspace 选项卡

选择 *Configure>Settings* 并单击 **Workspace** 选项卡以设置 Workspace 选项。

workspace（工作区）包含有关选定器件、调试工具和 / 或编程器、打开的窗口及其位置以及其他 IDE 配置设置的信息。

选项	说明
Automatically save workspace upon closing （关闭时自动保存工作区）	要在关闭 MPLAB IDE 时自动保存当前工作区的内容，请选择 Yes （是）。否则，单击 No （否）。若需要提示保存内容，请选择 Prompt （提示）。保存工作区的同时也会保存项目信息。
Reload last workspace at startup （在启动时重新加载上一次的工作区）	在启动 MPLAB IDE 时重新加载上一次工作区的设置。这也将重新加载上一次打开的项目。
Recent files list contains （列出的最近文件数）	指定在 *File>Recent Files* 下的列表中包含的最近使用过的文件的数量。
Recent workspaces list con-tains （列出的最近工作区数）	指定在 *File>Recent Workspaces* 下的列表中包含的最近使用过的工作区的数量。

9.22.2 Debugger 选项卡

选择 *Configure>Settings* 并单击 **Debugger** （调试器）选项卡以设置调试选项。

选项	说明
Automatically save files before running （在运行前自动保存文件）	确定在运行所选择的调试器之前保存了所有项目文件。
Browse for source if file is not found （若未找到文件请查找源文件）	如果找不到源文件，请使用 MPLAB IDE 已知的路径搜索源文件。
Show disassembly if source is unavailable （若找不到源文件显示反汇编代码）	如果找不到源文件，打开反汇编窗口。
Remove breakpoints upon importing a file （在导入文件时删除断点）	在导入文件时，即，使用 *File>Import* 添加调试 /hex 文件时，清除所有断点。

9.22.3 Program Loading 选项卡

选择 *Configure>Settings* 并单击 **Program Loading** 选项卡以设置在程序装载时是清零存储器还是不清零存储器。程序装载的意思就是编译、 make 或导入程序。

何时清零存储器

选项	说明
Clear memory before building a project （在编译项目之前清零存储器）	在编译项目之前清零所有存储器。
Clear memory after successfully building a project （在成功编译项目之后清零存储器）	在成功编译项目 （没有错误）之后清零所有存储器。

要清零的存储器

选项	说明
Clear program memory upon loading a program （在装载程序时清零程序存储器）	在装载程序时清零 / 不清零程序存储器。在使用并存项目时，请清除该复选框。
Clear configuration bits upon loading a program （在装载程序时清零配置位）	在装载程序时清零 / 不清零配置位的值。
Clear EE data upon loading a program （在装载程序时清零 **EE** 数据）	在装载程序时清零 / 不清零 EEPROM 数据存储器。
Clear user ID upon loading a program （在装载程序时清零用户 **ID**）	在装载程序时清零 / 不清零用户 ID 的值。

9.22.4 Hot Keys 选项卡

选择 *Configure>Settings* 并单击 **Hot Keys** 选项卡以便为各种 IDE 操作设置热键功能。热键 （也叫快捷键）为 IDE 菜单或工具栏命令的键盘快捷方式。

功能	说明
Hot Key mapping scheme （热键映射配置）	使用默认设置，或者使用 **Save As** 保存个人设置。会将您的自定义设置保存为一个扩展名为 `.hot` 的文件。
Command （命令）	选择想要为其指定热键的命令或者查看当前为该命令选择的热键。
Hot Key combination （热键组合）	为所选择的命令输入热键或者查看当前为该命令选择的热键。如果单击了 **Delete** 键或 **Backspace** 键，则将清空此字段并显示 “无”。当输入热键时，将在此处显示该键的值。 **注意：**必须敲击想要使用的真正的键，不要输入该键的名称 / 功能。例如，如果要为命令设置一个 **Ctrl + F8** 的热键组合，请在按住 **Ctrl** 键的同时按下 **F8** 键。
Hot Key currently in use by （当前使用的热键所对应的命令）	如果选择了已被使用的热键，此处就会显示出相应的命令。

9.22.4.1 有效热键

以下键不是有效的热键：

- Enter
- Backspace
- Delete
- Tab
- 任何字母或数字键以及 shift 和以上键的组合

如果选择了一个无效的键，在 Hot Key combination 中会显示 None 以反映该选择无效。

9.22.4.2 热键的多种分配

热键可能会被分配给多条命令。如果出现冲突，热键将应用于在表中首先遇到的 “已加载”的命令。

例如，热键被分配给一条 MPLAB ICD 2 调试器命令和一条 MPLAB PM3 编程器命令。但是，在分配热键时没有使能 MPLAB ICD 2，但使能了 MPLAB PM3。因此，热键仅应用于 MPLAB PM3 命令。如果同时使能了 MPLAB ICD 2 和 MPLAB PM3，热键只会应用于首先出现在热键表中的命令。

> **注：** 给命令分配热键时，请不要删除任何以前对该热键做出的命令分配。

9.22.5 Projects 选项卡

选择 *Configure>Settings* 并单击 **Projects** 选项卡来设置项目选项。

Project 包含编译应用程序所需要的文件 （源代码，链接描述文件等）以及与它们相关的各种编译工具和编译选项。

选项	说明
Close open source files on project close （在关闭项目时关闭打开的源文件）	在关闭项目时，关闭所有相关的已打开的源文件。
Clear output window before build （在编译项目之前清空输出窗口）	清空输出窗口的内容，然后进行编译。
Save project before build （在编译之前保存项目）	在编译之前保存项目的当前设置。
Save files before build （在编译之前保存文件）	如需在编译之前保存当前项目中的所有文件，请选择 Yes。否则，请选择 No。如果需要提示保存文件，请选择 Prompt。
Halt build on first failure （第一次失败时暂停编译）	在发生第一次错误时暂停编译。（对于一个失败的源文件可能会报告出多处错误。）
Use one-to-one project-workspace model （使用一对一的项目—工作区模型）	工作区中只允许有一个项目。欲知更多有关单项目和多项目工作区的信息，请参阅第 4 章 “项目和工作区”。

9.23 TABLE SETUP 对话框

右击 Watch 窗口并选择 “Import Table”或 “Export Table”以打开 Table Setup 对话框。从表格文件导入 Watch Value 数据或将 Watch Value 数据导出到表格文件。将 SFR/Symbo 值导出到一个文件，然后根据需要将它们重新导回。

相关的窗口 / 对话框是

- 第 **8.12** 节 **“Watch 窗口”**
- 第 **9.3** 节 **“Add Watch 对话框”**
- 第 **9.26** 节 **“Watch 对话框”**

操作	对话框	说明
导入 / 导出	**Start Address**	为下面选择的存储器类型输入起始地址。默认为在 Watch 窗口中所选择的项的地址。
导出	**End Address**	为下面选择的存储器类型输入结束地址。默认为在 Watch 窗口中所选择的项的地址。
导入 / 导出	**Memory**	选择要导入 / 导出的存储器类型。单击时，默认选中所有范围。
导入 / 导出	**Symbol Start Address**	如果可用的话，以符号表示所选的起始地址。
导出	**Single Column Output** **（单列输出）**	将输出内容以单列格式写入。
导入 / 导出	**Type Formatted** **（格式化输入）**	Import use （导入使用）：将一个电子表格列导入到实型数组。 Export use（导出使用）：将浮点 / 十进制数据移动到电子表格或绘图程序中。

9.24 USER ID MEMORY 对话框

某些 PICmicro MCU 和 dsPIC DSC 器件具有可存储校验和或其他标识号的存储单元。在编程 / 校验的过程中，可读写这些单元。根据器件的不同，在执行普通的 `TBLRD` 和 `TBLWT` 指令时，也可能允许访问这些单元。

选择 *Configure>ID Memory* 以打开 User ID Memory 对话框。

对话框	说明
Use Unprotected Checksum **（使用未受保护的校验和）**	使用未受保护的 （即代码保护关闭）校验和作为用户 ID。参见相应器件的编程规范以查看计算该校验和的方法。 **注意：**必须单击 **OK** 然后再次打开该对话框，才能看到未受保护的校验和已作为了用户 ID。并且，在进行 make 或编译之后此值还会改变。

9.25 VERSION-CONTROL 对话框

选择 Project>Version Control 以打开 Version-Control （版本控制）对话框。设置 MPLAB IDE 项目使之使用版本控制系统 （VCS）。

- Version-Control System——从下拉列表中选择可用的版本控制系统。
- System——如果需要的话，设置必要的参数以使用所选的 VCS。
- For Project “*project*.mcp”（关于项目 “*project*.mcp”）—— 为项目 *project*.mcp 设置必要的项目信息。

欲知更多有关版本控制系统的信息，请参阅**第 4.6 节 “使用版本控制系统 （VCS）”**。

Microsoft Visual SourceSafe

For Project “*project*.mcp”（关于项目 “*project*.mcp”）	说明
SRCSAFE.INI file（SRCSAFE.INI 文件）	为将要使用的数据库，指定 Visual SourceSafe 初始化文件的路径及其名称，例如 `C:\VSS\VSS_6.0\win32\scrsafe.ini`。单击 **Browse** 找到该文件。
VSS project path（VSS 项目路径）	在 Visual SourceSafe 内部指定项目文件的路径，例如 `$/Project Files/Project2`。

PVCS

System	说明
EXE directory （EXE 目录）	输入或浏览到包含 PVSC 可执行文件的目录。
Java directory （Java 目录）	输入或浏览到包含 PVSC java 脚本的目录。
Remember user name and password（记住用户名和密码）	选中该复选框，即可在第一次输入用户名和密码之后自动访问 PVSC 系统。
For Project “*project*.mcp”	**说明**
Database directory （数据库目录）	输入或浏览到包含 PVSC 数据库的目录，项目文件就来自此目录。
Archive directory （归档目录）	输入或浏览到包含 PVSC 归档文件的目录，项目文件也可能被保存在此目录中。

CVS

System	说明
CVS Executable （CVS 可执行文件）	输入或浏览到包含 CVS 可执行文件的目录。
Remember user name and password* （记住用户名和密码 *）	选中该复选框，即可在第一次输入用户名和密码之后自动访问 CVS 系统。
For Project “*project*.mcp”	**说明**
Host （主机）	输入作为 CVS 服务器的计算机的主机名或 IP 地址。这应该与在 `cvs login` 命令中给出的主机名或 IP 地址相符。
Port （端口）	输入 CVS 服务器上 CVS 服务的端口号。如果该服务器使用默认的端口值 2401，就可保留此字段为空。
Root （根节点）	CVS 服务器可以托管多个根节点，根节点的名称以前面带有正斜杠的形式提供。
Module （模块）	输入真正的 CVS 模块或给定 CVS 根目录中的顶层目录。

* 第一次在 MPLAB IDE 中使用 CVS 时，必须在 MPLAB IDE 之外登录 CVS，即

```
$ cvs -d :pserver:MyUserName@ServerHostName:/CVSRoot login
Logging in to :pserver:MyUserName@ServerHostName:2401:/CVSRoot
CVS password: ********
```

一旦登录完成，就可以在 MPLAB IDE 中使用 CVS 了。

9.26 WATCH 对话框

通过在 Watch 窗口中右击并选择 Properties 来设置 Watch 窗口。Watch 对话框中具有以下选项卡：

- Watch Properties 选项卡
- Preferences 选项卡
- General 选项卡 （参见第 **9.17.2** 节 **“General** 选项卡**”**）。

相关的窗口 / 对话框为：

- 第 **8.12** 节 **“Watch** 窗口**”**
- 第 **9.3** 节 **“Add Watch** 对话框**”**
- 第 **9.23** 节 **“Table Setup** 对话框**”**

9.26.1 Watch Properties 选项卡

此选项卡用于为 Watch 窗口中特定的符号设置属性。

属性	说明
Symbol （符号）	选择要为其设置属性的观察符号。选择在窗口中当前选中并显示的符号。
Size	选择选定的观察变量的值的位数。根据不同的器件，可选择 8、 16、 24、 32、 40 或 64 位。
Format	选择选定的观察变量的值的格式。有以下格式可供选择： - Hex （adecimal）（ 十六进制） - Binary （二进制） - Decimal （十进制） - ASCII - MCHP Float （MCHP 浮点） - IEEE Float （IEEE 浮点） - Single Bit （单个位） - dsPIC Integer （dsPIC 整数） - dsPIC Fractional（dsPIC 小数） - Double （双精度实型） **注意：**有了单个位， Symbol Name 中就不会显示出有关将显示哪个位的信息。
Byte Order （字节顺序）	查看或更改所选的观察符号的字节顺序。
Memory	查看所选的观察符号的存储器类型。

9.26.2 Preferences 选项卡

本选项卡用于为添加到 Watch 窗口中的新符号选择默认设置。

设置	说明
Float Format （浮点格式）	选择浮点格式。 • Use COFF float type （使用 COFF 浮点类型）——使用 COFF 文件中的信息。**这是默认设置。** • IEEE 754 32-bit （IEEE 754 32 位）——用于 -D32x 格式的 MPLAB C18 （v2.40 和更新的版本）、 MPLAB C30 和 HI-TECH C 。 • IEEE Modified 24-bit （IEEE 改进 24 位）——用于 -D24x 格式的 HI-TECH C 编译器。 • Microchip High:Low （Microchip 从高到低）——用于版本早于 v2.40 的 MPLAB C18 编译器。 • Microchip Low:High （Microchip 从低到高）——用于 CCS 编译器。 选定的浮点值还可用于定义编辑器中鼠标悬停区域的格式。
Unknown Type Format （未知类型的格式）	选择未知类型的格式。在不能指定类型时使用此格式。可从 Hex(adecimal) （十六进制）、Binary （二进制）、 Decimal （十进制）或 Char （字符）中选择。还要选择字节的顺序 （High:Low （从高到低）或 Low:High （从低到高））。选定的格式将用于 SFR、绝对地址和 ASM 符号。



第 10 章 MPLAB IDE 操作参考

10.1 简介

在这里讨论有关 MPLAB IDE 操作的参考信息。

- 命令行选项
- MPLAB IDE 所使用的文件
- 保存的信息

10.2 命令行选项

可以通过如下的命令行接口启动 MPLAB IDE：

`mplab [file] [/option]`

file = workspace.mcw

在 MPLAB IDE 中打开工作区 workspace.mcw。任何包含在该工作区中的项目也都将被打开。

option = nosplash （不闪屏）

程序启动时不显示闪屏

10.3 MPLAB IDE 所使用的文件

MPLAB IDE 文件的默认扩展名列于表 10-1 中。

表 10-1： MPLAB IDE 文件默认扩展名

扩展名	定义
a	存档 （库）文件——MPLAB LIB30
asm	汇编语言源文件——MPASM 汇编器
c	C 源文件
chm	已编译的 HTML 帮助文件
cod	包含符号信息和目标代码的文件
cof	包含符号信息和目标代码的文件
err	汇编器 / 编译器生成的错误文件
evt	事件文件——MPLAB ICE 2000
exe	程序 / 实用程序文件
fsti	文件激励文件——适用于 PIC17 MCU 的 MPLAB SIM
gld	链接描述文件－ MPLAB LINK30
h	C 头文件
hex	十六进制格式的机器码文件
inc	汇编语言头文件
lib	库文件——MPLIB 库管理器
lkr	链接描述文件－ MPLINK 链接器
lst	汇编器 / 编译器生成的完全列表文件
map	链接器生成的映射文件
mch	导出的数据文件

表 10-1：　MPLAB IDE 文件默认扩展名 （续）

扩展名	定义
mcp	项目信息文件
mps	Make 的编译状态文件
mcw	工作区信息文件
o	目标文件
psti	引脚激励文件——适用于 PIC17 MCU 的 MPLAB SIM
rsti	寄存器激励文件——适用于 PIC17 MCU 的 MPLAB SIM
ssti	同步激励文件——适用于 PIC17 MCU 的 MPLAB SIM
s	汇编语言源文件——MPLAB ASM30
sbs	SCL 发生器工作簿文件——MPLAB SIM
scl	来自 SCL 发生器的 SCL 文件——MPLAB SIM
stc	激励情形文件——MPLAB SIM
trc	跟踪保存文件
trg	触发文件——MPLAB ICE 2000
xrf	交叉汇编文件——MPASM 汇编器

10.4 保存的信息

有关 MPLAB IDE 设置的信息保存如下。

工作区

工作区包含以下信息：

- 选定的器件、调试工具和 / 或编程器。
- 调试工具 / 编程器设置信息。
- *Configure>Settings*， **Program Loading** 选项卡信息。
- 配置位设置。
- 打开的 IDE 窗口和它们的位置。
- 其他 IDE 系统设置。

此信息保存在 `.mcw` 文件中。

项目

项目包含以下信息：

- 编译某一应用程序所需的一组文件。
- 各种编译工具的相关文件。
- 编译选项。

此信息保存在 `.mcp` 文件中。

注册表

下列信息保存在 Windows 操作系统的注册表文件中。

- 语言工具名称和它们的安装位置。
- *Configure>Settings*， **Workspace** 选项卡上的大多数项。
- *Configure>Settings*， **Projects** 选项卡上的所有项。

INI 文件

以下信息保存在初始化 （`.ini`）文件中：

- 编辑器设置保存在 `mpeditor.ini` 文件中。
- 各窗口的状态信息保存在 `mplab.ini` 文件中。



第 3 部分——MPLAB 编辑器

注：



第 11 章 使用编辑器

11.1 简介

MPLAB 编辑器是 MPLAB IDE 集成开发环境中的集成部件。在 MPLAB IDE 运行时总是可以使用这个编辑器。

设计 MPLAB IDE 和 MPLAB 编辑器是为了让开发者能够轻松快捷地为 Microchip Technology 的 PICmicro 单片机（MCU）和 dsPIC 数字信号控制器（DSC）产品系列开发和调试固件。如需有关这些器件的更多信息，请访问 Microchip 网站。

编辑器在 MPLAB IDE 的文件窗口中运行。可使用菜单项 （File 和 Edit 菜单以及鼠标右键菜单）对编辑器进行控制。

MPLAB 编辑器允许使用以下功能进行一般的文本编辑。

11.1.1 插入、选择和删除文本

可以在插入或改写模式下输入文本。MPLAB 编辑器会在状态栏上以 “INS”或 “OVR”显示输入模式。输入的文本可以包含标准的字母数字和标点符号的任何组合。还可以使用特殊的键和 / 或键组合输入其他字符。

文本选择功能允许选择字符、单词、整行或其他文本块。可以复制或删除任何选定的文本。MPLAB 编辑器内建的查找和替换功能允许搜索并替换文本。

11.1.2 缩进文本和取消缩进

可以更改文本中一行或多行的缩进级别。

可以指定是否自动缩进新行。

11.1.3 为源代码模块定界

可以找到匹配的括号，如花括号和圆括号，这样做通常可以为文本段或程序源代码段定界。

11.1.4 撤销编辑操作

可以使用 Undo （撤消）命令取消编辑操作，而使用 Redo （ 重做）命令重复编辑操作。

11.2 配置编辑器

可以使用两个对话框配置编辑器：

- Editor Options 对话框
- Editor Color Customization 对话框

11.2.1 Editor Options 对话框

可以在 Editor Options 对话框中设置编辑器属性。选择 *Edit>Properties* 或在一个文件（编辑器）窗口中右击并选择 Properties。

- Editor 选项卡
- Text 选项卡
- Sizes 选项卡

11.2.1.1 EDITOR 选项卡

使用 Editor Options 对话框上的 Editor 选项卡设置编辑器属性，如表 11-1 所示。

表 11-1： EDITOR OPTIONS 对话框

选项	说明
Line Numbers	若选中，将显示出行号。
Print Line Numbers （打印行号）	若选中，将打印出行号。
Enable Color Printing （启用彩色打印）	若选中，会以彩色打印。这有助于保留语法信息。
Word Wrap （词换行）	若选中，会启用词换行。
Auto Indent （自动缩进）	若选中，新行会以与上一行相同数量的空格或制表符缩进。
Protect Read Only Files （保护只读文件）	若选中，则不能编辑只读文件。若清除，则能够编辑只读文件并将结果保存到另一个文件。
Enable Variable Mouseover Values （启用鼠标悬停时显示变量值）	若选中，在变量上悬停鼠标时会弹出变量的值。
Show Address in Mouseover （在鼠标悬停弹出的值中显示地址）	若选中，则会在鼠标悬停值前加上地址，例如 “Addr = 0x0003 Name = 0x00”
Double-click Toggles Breakpoint （双击交替启用 / 禁止断点）	若选中，双击某行会交替启用 / 禁止该行上的断点。若清除，双击某行则会选择该行的文本。
Find Wrap MessageBox （显示查找反卷消息框）	若选中，那么当查找反卷到顶部或尾部时，除了在状态栏的位置区显示 “Find Wrapped”（查找已反卷）以外，还会另外显示一个消息框。
Use Tabbed Window （使用带标签的窗口）	若选中，则会在单个窗口中显示多个标签，而不是为每个文件分别显示一个窗口。

11.2.1.2 TEXT 选项卡

使用 Editor Options 对话框的 Text 选项卡设置文本属性。

11.2.1.2.1 Fonts

- **Select Font**——单击选择编辑器字体。

仅适用于使能 Unicode 的操作系统。在 **National Language Code Page:** （国家语言代码页：）域，选择从 Unicode 文本转换到 ANSI 文本时将使用的值。

11.2.1.2.2 Colors

- **Choose Colors**——单击在 Editor Color Customization 对话框中选择上下文相关的颜色。以决定 “文本方式” 下的上下文颜色。
- **Default Colors** ——单击恢复默认的上下文相关的颜色。
- **User Defined Color File** （用户定义的颜色文件）——可以将颜色应用到在文件中定义的关键字上。按下面指定的规则创建文件，然后在该对话框的这一选项卡中输入或找到该文件。单击 **Choose Colors** 为选定的文件关键字指定颜色。
  - 文件的格式应该是每行一个关键字，每个关键字后跟一个回车符。
  - 文件中的关键字必须由字母和数字组成，并且第一个字符必须是字母。使用其他字符的关键字将被忽略。
  - 关键字应该是惟一的。如果选择了一个已经定义过的伪指令或保留的关键字，那么将无法确定要显示的颜色。

11.2.1.3 SIZES 选项卡

使用 Editor Options 对话框上的 Sizes 选项卡设置大小属性。

选项	说明
Tab Size （制表符宽度）	输入 1 到 16 之间的值，指定制表符的宽度。 **Insert spaces** （插入空格）——若选中该选项，则在文本中输入制表符时会在下一个制表符位置之前插入空格。 **Keep tabs** （保留制表符）——若选中该选项，则在文本中输入制表符时，会插入一个硬制表符。
Gutter Size （灰色区域大小）	输入 3 到 12 之间的值，指定窗口中灰色区域的宽度 （如果窗口中有灰色区域）。
Default Window Size （默认窗口大小）	输入编辑器窗口默认使用的大小值，该值为水平显示的字符数。

11.2.2 Editor Color Customization 对话框

可以在 Editor Options 对话框的 **Text** 选项卡上设置编辑器使用的颜色选项。所选颜色将被用于随后打开的所有语法类型与当前活动的源文件相同的文件。

颜色更改对话框包含一个特定符号类型的列表，这些类型的符号具有各自不同的颜色。每种符号类型的含义取决于应用颜色的文件类型。选中每种符号类型后，“Foreground Color”（前景色）和 “Background Color”（背景色）按钮就会显示该类型的当前颜色。若要更改颜色，可以单击这两个按钮，从而显示出 “颜色” 对话框以允许指定新的颜色。还可以将符号类型指定为以黑体和 / 或斜体形式出现，只需选中相应复选框即可。若想在不关闭对话框的情况下查看当前更改的效果，请单击**应用**。或者也可以单击**确定**使更改生效，但这样做会关闭对话框。

语法着色将标识出文本属于以下哪种类别。源文件的语法类型决定了某段文本属于哪种类别。

- ALL Settings （所有设置）——适用于所有类别的全局颜色设置
- Whitespace （空白）—— “不可见” 字符，比如空格和制表符
- Special Symbols （特殊符号）——标点符号和操作符
- Floating-Point Numbers （浮点数）
- Implicit-Base Numbers （默认数基数）——文件中以默认数基显示的数
- Binary Integers （二进制整数）
- Octal Integers （八进制整数）

- Decimal Integers （十进制整数）
- Hexadecimal Integers （十六进制整数）
- String Constants （字符串常量）
- Character Constants （字符常量）
- Preprocessor Directives （预处理伪指令）
- Trigraphs and Digraphs （三键组合字符和两键组合字符）——用于表示键盘上没有的特殊符号的按键组合 （过去在 C 语言中曾经使用过的功能）
- Comments （注释）
- Labels （标号）
- Reserved Words （保留字）
- Bad （错误）——无效字符
- Editor Window （编辑器窗口）——不属于上述任何类别的文本
- User Defined File （用户定义的文件）——由在 Text 选项卡上指定的用户文件所定义的文本

11.3 文件处理

文件处理时可以使用下列编辑器功能：

- 创建新文件
- 打开文件
- 打印文件
- 保存文件
- 关闭文件
- 语法类型
- 使用书签

11.3.1 创建新文件

要创建新文件：

1. 请单击 New File 图标，或从 File 菜单选择 New，还可以按 <CTRL> + <N> 键。
2. 会出现一个名为 “Untitled”的新窗口。这就是新建的文件。要将其命名为所需的名称，请选择 *File>SaveAs*。

11.3.2 打开文件

要打开现有的文件：

1. 有两种选择打开现有文件的方法。
   - 单击 Open File 图标，或从 File 菜单选择 Open ，还可以按 <CTRL> + <O> 键。将出现 “打开”对话框。在该对话框中，浏览到希望打开的文件的位置，并选择该文件。单击 Open 按钮。
   - 双击项目树中的文件以将其打开。
2. 选定的文件会显示在它自己的编辑器窗口中。如果选定文件已经打开，那么它当前的编辑器窗口将变为活动窗口。

11.3.3 打印文件

要打印已打开的文件：

1. 确保包含想要打印的文件的窗口为活动窗口。
2. 请单击 Print 图标，或从 File 菜单选择 Print，还可以按 <CTRL> + <P> 键。将显示 “打印”对话框。
3. 选择所需的打印选项，单击 “确定”。文件即可在所选的打印机上打印。

11.3.4 保存文件

要保存文件：

1. 确保包含想要保存的文件的窗口是活动窗口。
2. 请单击 Save 图标，或从 File 菜单选择 Save，还可以按 <CTRL> + <S> 键。将使用窗口上的名称保存文件。

要使用不同名称保存文件：

1. 确保包含想要保存的文件的窗口是活动窗口。
2. 从 File 菜单选择 Save As。将显示 "另存为"对话框。
3. 在该对话框中，浏览到想要保存文件的文件夹位置。
4. 若想使用其他名称保存文件，请在 "文件名"域中修改文件名。
5. 单击**保存**。

11.3.5 关闭文件

关闭文件有多种方法，如下所示：

- 通过 File 菜单：
  - 确保包含想要关闭的文件的窗口是活动窗口。
  - 从 File 菜单选择 Close。如果文件自上次保存后又被更改过了，则会提示您保存更改。
- 从文件窗口的系统按钮中选择 "关闭"。
- 单击文件窗口上的关闭按钮。
- 当想要关闭的文件位于活动窗口中时，可按 <CTRL> + <F4> 键关闭文件。

11.3.6 语法类型

为了进行语法着色， MPLAB 编辑器能够识别各种汇编器和编译器的语法。可使用以下方法指定源文件包含的语言类型。

1. 右击想为其设置语法类型的文件。
2. 从编辑器窗口的右键菜单中选择 Text Mode。将会看到一个菜单，其中包含编辑器当前识别的语法类型。
   - Disabled——无文本格式。
   - PIC16C5x Asm——12 位内核 MCU 器件
   - PIC16Cxxx Asm——14 位内核 MCU 器件
   - PIC17Cxxx Asm——16 位内核 MCU 器件
   - PIC18Cxxx Asm——增强型 16 位内核 MCU 器件
   - dsPIC Asm——dsPIC30F DSC 器件
   - Asm Listing （Asm 清单）——汇编程序清单
   - C——C 语言清单
   - Basic——Basic 清单
   - SCL——软件模拟器控制语言清单
3. 为您的文件选择最匹配的语言类型。

11.3.7 使用书签

从上下文 （鼠标右键）菜单中选择 Bookmarks。使用书签可以轻松地找到文件中的特定行。

- Toggle Bookmark （交替书签）——设置 / 除去当前选定行上的未命名书签。
- Next/Previous （下一个 / 上一个）——转至下一个 / 上一个书签。
- Clear All （清除所有）——清除所有书签

11.4 文本处理

文本处理时可以使用下列编辑器功能：

- 选择文本
- 转到指定行
- 剪切 / 删除文本
- 复制文本
- 粘贴文本
- 查找文本
- 替换文本
- 匹配括号
- 撤销编辑器操作
- 重做编辑器操作
- 缩进和取消缩进
- 格式化文本

11.4.1 选择文本

要选择一段字符：

- 使用鼠标：
  - 在要选择的文本一端单击鼠标。
  - 按住鼠标左键，将鼠标指针向文本另一端移动。
  - 松开鼠标左键。所选择的文本现在会以高亮显示。
- 使用键盘：
  - 使用导航键，将文本光标移动到想要选择的文本的一端。
  - 按下 shift 键，将文本光标移动到文本的另一端。
  - 松开 shift 键。所选择的文本现在会以高亮显示。

要选择整个词或整行：

注：	只有在编辑器选项中未选中 “Double-click Toggles Breakpoint”（双击交替启用 / 禁止断点）才可以使用这个功能。

- 词：双击想要选择的词。则会以高亮显示该词。
- 行：双击想要选择的行中的某个词。然后再次单击。则会以高亮显示这一行。

要选择整个文件的内容：

按 <CTRL> + <A> 键或从 Edit 菜单选择 Select All 。则会以高亮显示整个文件。

注：	如果在选择文本之后移动了光标，则将不再保持文本为选中状态。

11.4.2 转到指定行

无论光标位于文件中的哪个位置，都可以快速将其转到指定的任意行。

要转到活动窗口中的指定位置：

1. 从 Edit 菜单选择 Goto Line，或按 <CTRL> + <G> 键。将显示 Goto Line 对话框。
2. 在对话框中输入行号，然后单击 **OK** 按钮。编辑器会将光标转到指定的行，如果必要的话，还会滚动屏幕以显示这一行。

注：	如果输入的行号大于文件的总行数，则光标会转到文件的最后一行。

11.4.3 剪切 / 删除文本

要剪切文本：

1. 选择要剪切的文本。
2. 单击 Cut 图标，或从 Edit 菜单选择 Cut ，还可按 <CTRL> + <X> 键或 <Shift> + <Delete> 键。

将从源文档中删除选定文本并移到 Windows 剪贴板上，以便粘贴到其他地方。

要删除文本：

1. 选择要删除的文本。
2. 从 Edit 菜单选择 Delete，或按 <Delete> 键。

将从源文档中删除选定文本。如果误删了某些内容，请立即执行 Undo。

要删除单个字符：

1. 要删除光标左边的字符，请按 <Backspace> 键。
2. 要删除光标后面的字符，请按 <Delete> 键。

11.4.4 复制文本

要复制文本：

1. 选择想要复制的文本。
2. 单击 Copy 图标，或从 Edit 菜单选择 Copy，还可按 <CTRL> + <C> 键或 <Shift> + <Insert> 键。

要复制整列文本：

1. 将光标放置在要复制的文本列一角。
2. 按住 Alt 键，然后用鼠标单击拖拽以选择文本列。
3. 单击 Copy 图标、从 Edit 菜单选择 Copy 、右击所选列然后选择 Copy，或按 <CTRL> + <C> 键或 <Shift> + <Insert> 键。

注：	如果高亮显示的列不准确，请确保光标与鼠标指针位于同一行，这样才能正确选择。

选定文本会被复制到 Windows 剪贴板上，以便将其复制到其他地方。

11.4.5 粘贴文本

可以在 MPLAB 编辑器中将 Windows 剪贴板中的任何文本粘贴到某个文件。可以粘贴从同一个文件或另一个应用程序复制的文本。 Windows 剪贴板中必须存在文本，才可进行粘贴。

要从 Windows 剪贴板粘贴文本：

1. 请将光标移动到要插入文本的位置。
2. 单击 Paste 图标、从 Edit 菜单选择 Paste，或者按 <CTRL> + <V> 键或 <Shift> + <Insert> 键。

注：	粘贴文本时不会从剪贴板中删除该文本，因此可多次粘贴同一文本。

11.4.6 查找文本

要查找文件中的文本：

1. 确保想要搜索的文本已在活动窗口中打开。
2. 从 Edit 菜单选择 Find，或按 <CTRL> + <F> 键。将显示 Find 对话框。
3. 如果在打开 Find 对话框之前已选择了文本，则会在 “Find What”字段中显示选定的文本。若要搜索的文本未显示在 “Find What”字段中，请输入想要查找的文本，或从下拉列表中选择前面查找过的文本串。
4. 在该对话框中选择搜索时需要使用的任何查找选项。这些选项包括：
   - Match whole word only
   - Match case
   - Direction （方向）——Up
   - Direction （方向）——Down
5. 单击 **Find Next**。编辑器将把光标移动到文件中下一个 （上一个）匹配的文本处。如果没有找到文本，编辑器会显示一条消息表示未找到相应文本。

要重复以前的查找：

按 <F3> 键，若 Find 对话框仍保持打开，则请单击 **Find Next**。要以相反的方向重复上一次查找，请按 <Shift> + <F3> 键。

11.4.7 替换文本

要查找并替换文件中的文本：

1. 确保想要编辑的文本已在活动窗口中打开。
2. 从 Edit 菜单，选择 Replace，或按 <CTRL> + <H> 键。
3. 如果在打开 Replace 对话框之前已选择了文本，则会在 "Find What" 字段中显示选定的文本。若要被替换的文本未显示在 "Find What" 字段中，请输入想要被替换的文本。
4. 在 Replace With 字段中输入替换文本。
5. 在对话框中选择搜索时需要使用的任何查找 / 替换选项。这些选项包括：
   - Match whole word only
   - Match case
6. 单击：
   - **Find Next** 仅执行查找
   - **Replace** 替换最近的匹配文本
   - **Replace All** 替换所有的匹配文本
   - **Cancel** 关闭该对话框，而不执行任何操作。

要重复替换操作：

1. 按 <F3> 键。
2. 如果 Find/Replace 对话框仍保持打开，请单击 **Replace**。

最近一次执行的查找 / 替换操作将被执行，包括所有选中的查找 / 替换选项。

11.4.8 匹配括号

要识别括在一起的一段文本：

1. 将光标移动到开始 / 结束符号上。
2. 按 <CTRL> + <M> 键、选择 *Edit>Match*，或右击编辑窗口并选择 *Advanced>Match*。光标会被移动到匹配的符号上。

开始 / 结束符号包括：[ ] { } ( ) < >

11.4.9 撤销编辑器操作

如果刚刚更改了文件，则可以撤销最近一次做出的更改效果。

- 从 Edit 菜单选择 Undo ，或按 <CTRL> + <Z> 键。

可以连续多次重复撤销操作。每次均会撤销上一次撤销的操作之前的操作。

11.4.10 重做编辑器操作

如果刚刚使用 Undo 功能撤销了更改，则可以重做更改。

- 从 Edit 菜单选择 Redo ，或按 <CTRL> + <Y> 键。

可以连续多次重复 Redo 操作。

11.4.11 缩进和取消缩进

可以使用 *Edit>Properties>Editor Options* 中的 **Tabs** 选项卡进行文本的缩进和取消缩进操作。也可使用 <Tab>/<Shift>+<Tab> 键或 Indent Block/Outdent Block 命令（在右键菜单中的 Advanced 菜单项中）。

11.4.11.1 缩进

将按 Editor Options 对话框中指定的方式插入制表符或空格。

单行

1. 将光标移动到想要缩进的行的起始处。
2. 确保编辑器处于插入模式
3. 对于想要缩进的行，每按一次 <Tab> 键就可缩进一级。

自动缩进

新行会自动按前面的行缩进。

多行

1. 选择想要缩进的多个行。
2. 对于想要缩进的多个行，每按一次 <Tab> 键就可缩进一级。

11.4.11.2 取消缩进

单行

1. 将光标移动到想要取消缩进的行的起始处。
2. 对于想要取消缩进的行，每按一次 <Shift> + <Tab> 键就可取消一级缩进。

多行

1. 选择想要取消缩进的多个行。
2. 对于想要取消缩进的多个行，每按一次 <Shift> + <Tab> 键就可取消一级缩进。

注：	不能从第 1 列开始向后取消一行缩进。

11.4.12 格式化文本

可以使用在上下文 （右击鼠标）菜单中选择的选项格式化文本。

- 右击菜单中的 Text Mode。根据选定的汇编器或编译器的语法以及器件来格式化所有文本。
- *Advanced>Format Text*。将选定文本全部转为大写或小写。
- Properties，打开 Editor Options 对话框。设置文件 （编辑器）窗口。

11.5 使用调试功能

在 MPLAB IDE 中选择了调试器之后，就可在编辑器窗口中使用若干代码调试功能了。

- 筛选跟踪
- 断点

11.5.1 筛选跟踪

可在编辑器窗口中设置一个简单的筛选跟踪。但是，这样做将不能启用调试工具的跟踪功能。而且选入和排除跟踪是互相排斥的，即设置其中一个就会取消另一个功能。

选入跟踪

可通过以下方式选择 （筛选）要被跟踪进跟踪缓冲器内的代码。（跟踪缓冲器的内容显示在 Trace 窗口中）：

- 在编辑器窗口中选择要添加到跟踪的代码
- 右击该窗口并从菜单中选择 “Add Filter-in Trace”

排除跟踪

可通过以下方式选择 （筛选）要从跟踪缓冲器中排除的代码，而仅将未选中的代码跟踪进跟踪缓冲器：

- 在编辑器窗口中选择要从跟踪排除的代码
- 右击该窗口并从中菜单选择 “Add Filter-out Trace”

删除跟踪

要将代码从跟踪中删除，请右击筛选代码并选择 “Remove Filter Trace”（删除筛选跟踪）。

要删除用于跟踪的多个代码段，请右击编辑器窗口并选择 “Remove All Filter Traces”（删除所有筛选跟踪）。

11.5.2 断点

断点可以在指定的行上停止代码的执行，使您能够观察寄存器、位和变量值以进行调试。

断点是 MPLAB IDE 调试工具的标准功能。欲知更多信息，请参见**第 3 章 “步骤和教程”**。

11.6 键盘功能

以下键被指定用于编辑器：

- 快捷键
- 移动和选择键
- 特殊字符键

11.6.1 快捷键

编辑器可以使用下列键盘快捷键。要更改快捷键的指定，请使用 *Configure>Settings* 的 **Hot Keys** 选项卡。

11.6.1.1 编辑

按键	结果
CTRL + Z	撤销上一次编辑
CTRL + Y	重做上一次编辑
CTRL + X, Shift + Delete	剪切选定文本
CTRL + C, CTRL + Insert	复制选定文本
CTRL + V, Shift + Insert	粘贴文本
Delete	删除选定文本
CTRL + A	选择窗口中的所有文本
CTRL + F	查找
CTRL + H	查找 / 替换
F3	向下重复查找 / 替换
SHIFT + F3	向上重复查找 / 替换
CTRL + G	转到指定行
CTRL + M	匹配括号

11.6.1.2 文件管理

按键	结果
CTRL + N	创建新文本文件
CTRL + O	打开现有文件
CTRL + F4	关闭现有文件
CTRL + S	保存当前文件
CTRL + P	打印当前文件

11.6.1.3 导航

按键	结果
CTRL + F6	下一个打开的窗口
SHIFT + CTRL + F6	上一个打开的窗口
CTRL + K	交替设置 / 删除书签
CTRL + L	下一个书签
F1	帮助

11.6.2 移动和选择键

在编辑文本时可以使用表 11-2 所示的键盘按键。

表 11-2： 编辑文本的按键

按键	结果
CTRL + A	选择文件中的所有文本
UP	将光标向上移动一行
SHIFT + UP	将光标向上移动一行，以扩展选择范围
DOWN	将光标向下移动一行
SHIFT + DOWN	将光标向下移动一行，以扩展选择范围
LEFT	将光标向左移动一个字符
SHIFT + LEFT	将光标向左移动一个字符，以扩展选择范围
CTRL + LEFT	将光标向左移动一个词
CTRL + SHIFT + LEFT	将光标向左移动一个词，以扩展选择范围
RIGHT	将光标向右移动一个字符
SHIFT + RIGHT	将光标向右移动一个字符，以扩展选择范围
CTRL + RIGHT	将光标向右移动一个词
CTRL + SHIFT + RIGHT	将光标向右移动一个词，以扩展选择范围
PGDN	将光标向下移动一页
SHIFT + PGDN	将光标向下移动一页，以扩展选择范围
PGUP	将光标向上移动一页
SHIFT + PGUP	将光标向上移动一页，以扩展选择范围
HOME	将光标移动到行首
SHIFT + HOME	将光标移动到行首，扩展选择范围
CTRL + HOME	将光标移动到文件首
CTRL + SHIFT + HOME	将光标移动到文件首，以扩展选择范围
END	将光标移动到行末
SHIFT + END	将光标移动到行末，以扩展选择范围
CTRL + END	将光标移动到文件末
CTRL + SHIFT + END	将光标移动到文件末，以扩展选择范围

11.6.3 特殊字符键

要输入任一字符：

1. 将键盘数字键区设为数字模式 （按 <NumLock> 键或按住 <Shift> 键）。
2. 按住 <Alt> 键。
3. 输入所需的十进制数字 （0 到 255）。
4. 松开 <ALT> 键。

如果输入的字符可见，则它将会显示在文件窗口中的光标处。

插入 / 改写模式：

使用 Insert 键可在 INS 和 OVR 模式之间切换，具体的模式将显示在状态栏中。

键	插入模式	改写模式
Enter	插入新行	移动到下一行的第一个字符 （如果有）
TAB 或 CTRL + I	插入制表符	移动到下一个制表符位置 （如果有）

11.7 编辑器疑难解答

本节旨在帮助您解答使用 MPLAB 编辑器过程中可能出现的任何问题、错误或难点。

11.7.1 常见问题 /FAQ

上下文菜单中的某些项被禁用了。

根据所选择的调试和 / 或编程器工具，某些项可能无法使用，因此它们会灰掉。

按下 <CTRL> + <M> 键后没有反应。

按下 <CTRL> + <M> 键时文本光标必须位于开始 / 结束符号上。如果符合上述条件，匹配操作就会将光标移动到与之匹配的符号上。如果没有匹配的符号，光标将不会移动。

开始 / 结束符号包括：[ ] { } ( ) < >

当我插入回车后，新行总是在第 1 列开始，当前设置的缩进级别不起作用。

必须在 Tab Settings 对话框中选中 Auto Indent，才能使用自动缩进功能。

11.7.2 怎样将更改保存到文件？

若自从上次将文件保存到磁盘之后，已经对文件做了修改。那么在 MPLAB 编辑器关闭文件之前，您可以选择更新磁盘副本。

命令	结果
有	将窗口的内容保存到磁盘。如果文件没有命名，则会提示对其命名。
否	关闭文件，放弃上一次保存之后所做的更改。
取消	取消关闭操作，返回窗口继续编辑。





第 4 部分——MPLAB SIM

注：



第 12 章 软件模拟器概述

12.1 简介

MPLAB SIM 是一个不连续事件软件模拟器，适用于：

- PICmicro 单片机 （MCU）系列
- dsPIC 数字信号控制器 （DSC）系列

已将软件模拟器集成到 MPLAB IDE 集成开发环境中。 MPLAB SIM 调试工具是专为模拟 Microchip 的 MCU 和 DSC 器件的运行而设计的，它可以辅助调试这些器件的软件。

如果不熟悉 MPLAB SIM，可以从 MPLAB IDE 教程开始学习。

- 软件模拟器功能
- 软件模拟器—— PICmicro MCU
- 软件模拟器模型——dsPIC DSC
- 软件模拟器执行

12.2 软件模拟器功能

MPLAB SIM 使您能够：

- 修改目标代码并立即重新执行目标代码
- 向模拟的处理器注入外部激励
- 设置引脚和寄存器在预先指定的时间间隔内的值
- 跟踪目标代码的执行 （仅用于 MCU）

12.3 软件模拟器—— PICmicro MCU

PICmicro MCU 采用与冯 • 诺依曼 （Von Neumann）结构 （程序存储器与数据存储器合二为一）相反的哈佛 （Harvard）结构 （程序存储器和数据存储器分离）。因此，程序存储器指令字长并不受 8 位数据长度的限制。根据程序存储器、内核或指令长度对 PICmicro MCU 进行分类。

在撰写本文档之际，已存在以下内核及其相关器件的型号：

- 12 位内核器件的型号——PIC12C5XX、PIC12CE5XX、PIC16X5X 和 PIC16C505
- 14 位内核器件的型号——PIC12C67X、 PIC12CE67X、 PIC12F629/675 和 PIC16
- 16 位内核器件的型号——PIC17
- 16 位内核器件型号——PIC18

12 位、 14 位和 16 位 （PIC17）内核的器件具有字寻址程序存储器空间。

16 位 （PIC18）内核的器件具有按字节组织的程序存储器空间。芯片对如何访问程序存储空间有所限制，特别是当对程序存储器使用长写操作时。软件模拟器可能并不会在所有的情况下显示相同的限制。请参阅数据手册，获取正确的操作。

12.3.1 12 位内核器件的型号——PIC12C5XX、PIC12CE5XX、PIC16X5X 和 PIC16C505

以下主题讨论可在软件模拟器中模拟的 12 位内核器件的功能。

- 12 位内核 I/O 引脚
- 12 位内核 CPU
- 12 位内核外设

12.3.1.1 12 位内核 I/O 引脚

12 位内核器件与其他外设复用 I/O 引脚，因此引脚有多个名字。 MPLAB SIM 只识别以下引脚名作为有效的 I/O 引脚：

> **注：** 只能使用在具体器件的数据手册中描述的那些引脚。

- MCLR
- T0CKI
- GP0-GP5
- RA0-RA3
- RB0-RB7
- RC0-RC7
- RD0-RD7
- RE4-RE7

12.3.1.2 12 位内核 CPU

复位状态

MPLAB SIM 软件模拟器支持所有的复位状态。

STATUS 寄存器中的超时 （Time-out， TO）位和掉电 （Power-down， PD）位反映了相应的复位状态。该功能对在代码中模拟产生各种上电和超时非常有用。

通过激励将 MCLR 驱动为低电平 （然后再驱动为高电平），或选择 *Debugger>Reset>MCLR Reset* 可以很容易地模拟正常运行中的 MCLR 复位。

看门狗定时器

MPLAB SIM 软件模拟器能完全模拟看门狗定时器。

WDT 的周期由 OPTION_REG 寄存器中的预 / 后分频比设置决定。基本周期 （当预 / 后分频比为 1:1 时）大致等于器件的数据手册中指令周期的倍数。

在 Configuration Bits 对话框 （*Configuration>Configuration Bits*）中使能 / 禁止 WDT。

在使能了 WDT 并正确设置了预 / 后分频比的情况下，模拟 WDT 超时，事实上 WDT 也将溢出。当 WDT 超时时，软件模拟器将根据在 Settings 对话框中 Break Options （中断选项）选项卡内的设置暂停或复位。

12.3.1.3 12 位内核外设

在支持内核的同时， MPLAB SIM 还完全支持 TIMER0 定时 / 计数器模块的内外两种时钟模式。

由于 MPLAB SIM 在指令周期边界处执行，因此不能模拟精度低于 1 个 TCY 的情况，切记这一点非常重要。

12.3.2 14 位内核器件的型号——PIC12C67X、 PIC12CE67X、 PIC12F629/675 和 PIC16

以下主题讨论可在软件模拟器中模拟的 14 位内核器件的功能。

- 14 位内核 I/O 引脚
- 14 位内核中断
- 14 位内核 CPU
- 14 位内核外设

12.3.2.1 14 位内核 I/O 引脚

14 位内核器件与其他外设复用 I/O 引脚，因此引脚有多个名字。 MPLAB SIM 只识别以下引脚名作为有效的 I/O 引脚：

注：	只能使用在具体器件的数据手册中描述的那些引脚。

- MCLR
- GP0-GP5
- RA0-RA5
- RB0-RB7
- RC0-RC7
- RD0-RD7
- RE0-RE7

12.3.2.2 14 位内核中断

支持以下中断：

- Timer0 溢出
- Timer1 溢出
- Timer2
- CCP1
- CCP2
- 端口 RB <7:4> 电平变化
- RB0/INT 引脚的外部中断
- 比较器
- A/D 转换完成
- EEPROM 写入完成

12.3.2.3 14 位内核 CPU

复位状态

MPLAB SIM 软件模拟器支持所有的复位状态。

STATUS 寄存器中的超时 （TO）位和掉电 （PD）位反映了相应的复位状态。此功能对在用户代码中模拟产生各种上电和超时非常有用。

通过激励将 MCLR 驱动为低电平 （然后再驱动为高电平），或选择 *Debugger>Reset>MCLR Reset* 可以很容易地模拟正常运行中的 MCLR 复位。

休眠

当执行 Sleep 指令时， PLAB SIM 将一直显示 “asleep”（休眠）直到产生从休眠状态唤醒的条件。例如，若使能了看门狗定时器，当它超时时 （取决于预 / 后分频比设置）就会将处理器从休眠中唤醒。

使用 Timer1 唤醒休眠状态下的处理器可以作为从休眠状态唤醒的一个实例。在该例中，当处理器处于休眠时， Timer1 仍继续递增直至溢出。如果允许中断，定时器溢出将唤醒处理器，并且处理器将转入中断向量处执行。

看门狗定时器

MPLAB SIM 软件模拟器能完全模拟看门狗定时器。

WDT 的周期由 OPTION_REG 寄存器中的预 / 后分频比设置决定。基本周期 （当预 / 后分频比为 1:1 时）大致等于器件的数据手册里指令周期的倍数。

在 Configuration Bits 对话框 （*Configuration>Configuration Bits*）中使能 / 禁止 WDT。

在使能了 WDT 并正确设置了预 / 后分频比的情况下，模拟 WDT 超时，事实上 WDT 也将溢出。当 WDT 超时时，软件模拟器将根据在 Settings 对话框中 Break Options 选项卡内的设置暂停或复位。

12.3.2.4 14 位内核外设

在提供内核支持的同时，除了支持通用 I/O 以外， MPLAB SIM 还支持下列外设模块。欲知有关采用何种符号的信息，请参阅您正在使用的特定器件的数据手册。

注：	即使软件模拟器不支持外设功能，仍可读 / 写相应的外设寄存器。

- 定时器
- CCP/ECCP
- 比较器 （受限）
- A/D 转换器 （受限）
- USART
- EEPROM 数据存储器

在所有外设上均可实现延时，但不能实现中断响应延时。

定时器

完全支持 *Timer0* （和 Timer0 溢出中断），并且 Timer0 按照内部或外部时钟递增计数。由于激励要求，时钟输入必须具有最少 1 个 TCY 的高电平周期和最少 1 个 TCY 的低电平周期。

支持 *Timer1* 的各种模式，但依靠外接晶振运行的计数器模式除外。 MPLAB SIM 支持 Timer1 溢出中断和唤醒中断。 MPLAB SIM 并不模拟 RC0/RC1 引脚上的外接振荡器，但可以给这些引脚分配时钟激励。

MPLAB SIM 完全支持 *Timer2* 及其溢出中断。

CCP/ECCP

捕捉

MPLAB SIM 完全支持捕捉和捕捉中断。

比较

MPLAB SIM 支持比较模式、比较中断及特殊事件触发 （由 CCP1 复位 Timer1）。

PWM

支持 PWM 输出 （仅限于分辨率大于 1 个 TCY 的情况）。

比较器 （受限）

MPLAB SIM 只模拟不使用参考电压的比较器模式。

A/D 转换器 （受限）

MPLAB SIM 可实现所有的寄存器、定时功能和中断的产生。但是，在 A/D 转换结束时并不向 A/D 结果寄存器 （ADRES）装入任何有效的值。

> **注：** 如果带有 A/D 转换器的处理器 （如 PIC16C74、PIC16F877 等）上的 I/O 引脚存在问题，请确定寄存器 ADCON 已把这些 I/O 引脚配置为数字 I/O 而非模拟输入。对于大多数处理器，在 ADCON （或 ADCON1）设置正确之前，这些引脚均被默认为模拟输入并且相关引脚不能用作 I/O 引脚。

USART

支持 USART 功能。

EEPROM 数据存储器

可完全模拟 EEPROM 数据存储器。可完全实现寄存器和读 / 写周期。写周期大约为 10 ms （接近于指令周期的倍数）。

软件模拟器模拟 EECON1 寄存器中的 WRERR 和 WREN 控制位的功能。

12.3.3 16 位内核器件的型号——PIC17

以下主题讨论了可在软件模拟器中模拟的 16 位内核器件的功能。

- 16 位内核 （PIC17） I/O 引脚
- 16 位内核 （PIC17）中断
- 16 位内核 （PIC17） CPU
- 16 位内核 （PIC17）处理器模式
- 16 位内核 （PIC17）外设

12.3.3.1 16 位内核 （PIC17） I/O 引脚

PIC17 器件与其他外设复用 I/O 引脚，因此引脚有多个名字。 MPLAB SIM 只识别以下引脚名作为有效的 I/O 引脚：

注：	只能使用在具体器件的数据手册中描述的那些引脚。

- MCLR
- RA0-RA5
- RB0-RB7
- RC0-RC7
- RD0-RD7
- RE0-RE2

12.3.3.2 16 位内核 （PIC17）中断

支持以下中断：

- INT 引脚上的外部中断
- TMR0 溢出中断
- RA0 引脚上的外部中断
- 端口 B 输入改变中断
- Timer/Counter1 （定时器 / 计数器 1）中断
- Timer/Counter2 （定时器 / 计数器 2）中断
- Timer/Counter3 （定时器 / 计数器 3）中断
- Capture1 （捕捉 1）中断
- Capture2 （捕捉 2）中断

12.3.3.3 16 位内核 （PIC17） CPU

复位状态

MPLAB SIM 软件模拟器支持所有的复位状态。

CPUSTA 寄存器中的超时 （TO）位和掉电 （PD）位反映了相应的复位状态。该功能对在代码中模拟产生各种上电和超时非常有用。

通过激励将 MCLR 驱动为低电平 （然后再驱动为高电平），可以很容易地模拟正常运行或休眠过程中的 MCLR 复位。

休眠

当执行 Sleep 指令时， MPLAB SIM 会一直显示 “asleep”直到产生从休眠状态唤醒的条件。例如，若使能了看门狗定时器，当它超时时 （取决于预 / 后分频比设置）就会将处理器从休眠中唤醒。

端口 B 上输入电平的变化可以作为一个从休眠状态唤醒的实例。如果允许中断并且 GLINTD 位置 1，那么处理器就会被唤醒并将从 Sleep 命令的下一条指令恢复指令执行。如果 GLINTD = 0，那么将会发生正常的中断响应。

看门狗定时器

MPLAB SIM 软件模拟器能完全模拟看门狗定时器。

WDT 周期取决于后分频比配置位 WDTPS0:1。基本周期 （当后分频比为 1:1 时）大致等于器件的数据手册里指令周期的倍数。将配置位 WDTPS0:1 置为 00 会禁止 WDT。

在 Configuration Bits 对话框 （*Configuration>Configuration Bits*）中使能 / 禁止 WDT 并设置预 / 后分频比。

在使能了 WDT 并正确设置了后分频比的情况下，模拟 WDT 超时，事实上 WDT 也将溢出。当 WDT 超时时，软件模拟器将根据在 Settings 对话框中 Break Options 选项卡内的设置暂停或复位。

12.3.3.4 16 位内核 （PIC17）处理器模式

MPLAB SIM 支持以下处理器模式：

- 单片机 （默认）
- 微处理器
- 带有引导程序的微处理器
- 扩展单片机

用代码 （`__config`）或在 Configuration Bits 对话框中设置处理器模式。

12.3.3.5 16 位内核 （PIC17）外设

在提供内核支持的同时，除了支持通用 I/O 以外， MPLAB SIM 还支持下列外设模块：

- Timer0
- Timer1 和 Timer2
- Timer3 和 捕捉
- PWM

在所有的外设上均可实现延时，但不能实现中断响应延时。

Timer0

完全支持 Timer0 和 Timer0 溢出中断，并且 Timer0 按照内部或外部时钟递增计数。还可模拟从外部时钟沿到定时器加 1 的延时和中断响应延时。由于激励文件要求，时钟输入必须具有最少 1 个 T_{CY} 的高电平周期和最少 1 个 T_{CY} 的低电平周期。

Timer1 和 Timer2

完全支持 Timer1 和 Timer2 的各种模式。可模拟从时钟沿到定时器加 1 （当配置为在外部时钟的上升沿或下降沿加 1）的延时及中断响应延时。由于激励文件要求，时钟输入必须具有最少 1 个 T_{CY} 的高电平周期和最少 1 个 T_{CY} 的低电平周期。

Timer3 和捕捉

MPLAB 软件模拟器完全支持所有模式下的 Timer3 和捕捉模块。完全支持从时钟沿到定时器加 1（当配置为外部模式）的延时、捕捉延时和中断响应延时。由于激励文件要求，时钟输入必须具有最少 1 个 TCY 的高电平周期和最少 1 个 TCY 的低电平周期。

PWM

支持所有两种模式的 PWM 输出（仅限于分辨率大于 1 个 TCY 的情况）。

12.3.4 16 位内核器件型号——PIC18

以下主题讨论了可在软件模拟器中模拟的增强型 16 位内核器件的功能。

- 16 位内核（PIC18） I/O 引脚
- 16 位内核（PIC18）中断
- 16 位内核（PIC18） CPU
- 16 位内核（PIC18）外设

12.3.4.1 16 位内核（PIC18） I/O 引脚

PIC18 器件与其他外设复用 I/O 引脚，因此引脚有多个名字。 MPLAB SIM 只识别以下引脚名作为有效的 I/O 引脚：

注：	只能使用在具体器件的数据手册中描述的那些引脚。

- MCLR
- RA0-RA5
- RB0-RB7
- RC0-RC7
- RD0-RD7
- RE0-RE2

12.3.4.2 16 位内核（PIC18）中断

支持以下中断：

- INT 引脚上的外部中断
- TMR0 溢出中断
- RA0 引脚上的外部中断
- 端口 B 输入改变中断
- Timer/Counter1（定时器 / 计数器 1）中断
- Timer/Counter2（定时器 / 计数器 2）中断
- Timer/Counter3（定时器 / 计数器 3）中断
- Capture1（捕捉 1）中断
- Capture2（捕捉 2）中断

12.3.4.3 16 位内核 （PIC18） CPU

复位状态

MPLAB SIM 软件模拟器支持所有的复位状态。

RCON 寄存器中的超时 （TO）位和掉电 （PD）位反映了相应的复位状态。该功能对在用户代码中模拟产生各种上电和超时非常有用。

不能使用激励控制翻转 MCLR 引脚电平，来实现 MCLR 复位。必须使用 *Debugger>Reset>MCLR Reset*。

休眠

当执行 SLEEP 指令时， MPLAB SIM 会一直显示 “asleep”直到产生从休眠状态唤醒的条件。例如，若使能了看门狗定时器，当它超时时 （取决于预 / 后分频比设置）就会将处理器从休眠中唤醒。

端口 B 输入电平变化中断可以作为一个从休眠状态唤醒的实例。如果允许中断（RBIE）且选择了中断优先级位 （RBIP），那么处理器会被唤醒，并将从 SLEEP 指令的下一条指令恢复指令执行。

看门狗定时器

MPLAB SIM 软件模拟器能完全模拟看门狗定时器。

WDT 周期取决于预 / 后分频比配置位 WDTPS0:2。基本周期 （当预 / 后分频比为 1:1 时）大致等于器件的数据手册里指令周期的倍数。

将配置位 WDTEN 置 0 将禁止 WDT，除非 WDT 已被 WDTCON 寄存器中的 SWDTEN 位使能。无论 SWDTEN 位的值为 1 还是 0，将配置位 WDTEN 置 1 总会使能 WDT。

在 Configuration Bits 对话框 （*Configuration>Configuration Bits*）中使能 / 禁止 WDT 并设置预 / 后分频比。

在使能了 WDT 并正确设置了预 / 后分频比的情况下，模拟 WDT 超时，事实上 WDT 也将溢出。当 WDT 超时时，软件模拟器将根据在 Settings 对话框中 Break Options 选项卡内的设置暂停或复位。

12.3.4.4 16 位内核 （PIC18）外设

在提供内核支持的同时，除了支持通用 I/O 以外， MPLAB SIM 还支持下列外设模块：

- 定时器
- CCP/ECCP
- 比较器 （受限）
- A/D 转换器 （受限）
- USART
- EEPROM 数据存储器

在所有的外设上均可实现延时，但不能实现中断响应延时。

定时器

完全支持 *Timer0* （和 Timer0 溢出中断），并且 Timer0 按照内部或外部时钟递增计数。由于激励要求，时钟输入必须具有最少 1 个 TCY 的高电平周期和最少 1 个 TCY 的低电平周期。

还支持各种模式下的*所有其他定时器*，但使用外接晶振的模式除外。 MPLAB SIM 支持定时器溢出中断和唤醒中断。虽然不能模拟外接振荡器，但可给这些引脚分配时钟激励。

CCP/ECCP

捕捉

MPLAB SIM 完全支持捕捉和捕捉中断。

比较

MPLAB SIM 支持比较模式、比较中断及特殊事件触发 （由 CCP 复位定时器）。

PWM

支持 PWM 输出 （仅限于分辨率大于 1 个 TCY 的情况）。

比较器 （受限）

MPLAB SIM 只模拟不使用参考电压的比较器模式。

A/D 转换器 （受限）

MPLAB SIM 可实现所有的寄存器、定时功能和中断的产生。但是，在 A/D 转换结束时并不向 A/D 结果寄存器 （ADRES）装入任何有效的值。

> **注：** 如果带有 A/D 转换器的处理器 （如 PIC16C74、PIC16F877 等）上的 I/O 引脚存在问题，请确定寄存器 ADCON 已把这些 I/O 引脚配置为数字 I/O 而非模拟输入。对于大多数处理器，在 ADCON （或 ADCON1）设置正确之前，这些引脚均被默认为模拟输入并且相关引脚不能用作 I/O 引脚。

USART

支持 USART 功能。

EEPROM 数据存储器

可完全模拟 EEPROM 数据存储器。可完全实现寄存器和读 / 写周期。写周期大约为 10 ms （接近于指令周期的倍数）。

软件模拟器模拟 EECON1 寄存器中的 WRERR 和 WREN 控制位的功能。

12.4 软件模拟器模型——dsPIC DSC

dsPIC 数字信号控制器 （DSC）融合了数字信号处理 （DSP）内核和 PICmicro 单片机 （MCU）的外设功能。 DSC 采用了改进的哈佛 （Harvard）结构以提供独立的程序存储器空间和 16 位的数据存储器空间。

以下主题讨论了可在软件模拟器中模拟的 dsPIC 器件的功能。

- I/O 引脚
- 异常 （陷阱 / 中断）
- 系统集成模块
- 存储器
- 外设

12.4.1 I/O 引脚

dsPIC 与其他外设复用 I/O 引脚，因此引脚有多个名字。软件模拟器只识别在标准器件头文件中指定的引脚名作为有效 I/O 引脚。因此，应参阅相应器件的头文件（`p`*`DeviceNumber`*`.inc`）以确定正确的引脚名。

12.4.2 异常 （陷阱 / 中断）

即使目前暂不支持某外设，软件模拟器仍可支持所有的内核和外设的陷阱及中断。

dsPIC 内核具有向量化异常处理结构，此结构可最多处理 8 个陷阱和 54 个中断 （即，总共 62 个独立向量）。每个中断均设有由用户分配的从 1 到 7 的优先级 （1 为最低优先级， 7 为最高优先级）。如果发生冲突 （两个中断优先级相同），中断服务会根据预定义的基于硬件的 “自然顺序”进行处理。

12.4.3 系统集成模块

复位源

MPLAB SIM 软件模拟器支持所有的复位源。

正如器件的数据手册所述的那样，在不同的复位状态下 RCON 寄存器中相应状态位的置 1 或清 0 也会有所不同。软件使用这些状态位来判断复位的性质。

通过激励将 MCLR 驱动为低电平 （然后再驱动为高电平），或选择 *Debugger>Reset>MCLR Reset* 可以很容易地模拟正常运行中或休眠 / 空闲状态下的 MCLR 复位。

休眠 / 空闲

当执行 PWRSAV 指令时， MPLAB SIM 会一直显示 “asleep”（休眠）或 “idle”（空闲）直到产生唤醒条件。例如，若使能了看门狗定时器，当它超时时 （取决于预 / 后分频比设置）就会将处理器从休眠中唤醒。

使用 Timer1 唤醒休眠状态下的处理器就是从休眠状态唤醒的一个实例。在该例中，当处理器处于休眠时， Timer1 仍继续递增直至与周期计数器匹配。如果允许中断，定时器将唤醒处理器，并且处理器将跳转到中断向量处执行。

看门狗定时器

MPLAB SIM 软件模拟器能完全模拟看门狗定时器。

通过 FWDT 配置寄存器中的配置位 （FWDTEN）可 “使能”或 “禁止”看门狗定时器。将 FWDTEN 置 1，使能看门狗定时器。将 FWDTEN 置 0 允许用户软件通过 SWDTEN （RCON<5>）控制位使能 / 禁止看门狗定时器。

看门狗定时器 （WDT）的周期取决于 FWDT 寄存器中的预 / 后分频比设置。基本周期（当预 / 后分频比为 1:1 时）大致等于器件的数据手册里指令周期的倍数。

在 Configuration Bits 对话框 （*Configuration>Configuration Bits*）中使能 / 禁止 WDT 并设置预 / 后分频比。

在使能了 WDT 并正确设置了预 / 后分频比的情况下，模拟 WDT 超时，事实上 WDT 也将溢出。当 WDT 超时时，软件模拟器将根据在 Settings 对话框中 Break Options 选项卡内的设置暂停或复位。

12.4.4 存储器

MPLAB SIM 可以模拟 dsPIC 器件存储器——闪存程序存储器和数据 RAM，但安全方面的功能除外。即不模拟受保护的存储器。

12.4.5 外设

MPLAB SIM 支持所有的外设模块，但在 Limitations （限制）中列出的功能模块除外。

在所有的外设上均可实现延时，但不能实现中断响应延时。

12.5 软件模拟器执行

以下主题详细说明了 MPLAB SIM 的操作。

- 执行速度
- 在指令周期边界上执行
- I/O 时序

12.5.1 执行速度

当 MPLAB SIM 实时模拟运行时，计算机以 CPU 允许的最快速度执行指令。但速度通常低于实际器件在额定时钟速度下的运行速度。

软件模拟器运行的速度取决于计算机速度和后台正在运行的其他任务数。软件模拟器必须更新所有的模拟寄存器和 RAM，它还必须监视 I/O、设置和清除标志，检查软件中的断点和跟踪点，以及使用在计算机 CPU 中执行的指令来模拟指令。

不连续事件软件模拟器的执行速度比面向硬件的解决方案小数个数量级。较慢的执行速度既可看作是一种阻碍也可看作是一种工具。软件模拟器试图提供尽可能快的模拟周期，它根据不同的工作模式可以工作在每指令几毫秒的数量级上。

注：	代码中经常使用循环以产生定时延迟。在使用软件模拟器时，您可以缩短这些延时或在条件允许下移除代码中带有 “IFDEF”语句的部分，以提高模拟速度。

一般来说，当在处于软件模拟器模式的情况，论及 “实时”，即表示软件模拟正在以计算机能模拟指令的最快速度执行模拟的 PICmicro MCU 代码。

12.5.2 在指令周期边界上执行

软件模拟器在指令周期边界上执行，因而不能模拟精度低于一个指令周期 （TCY）的情况。 MPLAB SIM 软件模拟器是一个不连续事件软件模拟器，用于在指令周期边界（或 TCY = 4*Tosc，其中 Tosc 为输入时钟周期）处，施加所有激励和产生所有的响应。因此，不能精确地模拟一些实际事件。这些事件可分为以下几类：

- 纯异步事件
- 周期小于一个指令周期的事件

指令边界模拟的最终结果就是所有的事件在指令边界处得到同步，所有小于一个指令周期的事件均不能被识别。

指令周期边界上的模拟影响下列功能和外设：

注：	目前软件模拟器并不支持所有在此列出的外设。

- 虽然定时器预分频器能接受小于一个周期的时钟脉冲输入，但 MPLAB SIM 却不能模拟小于一个周期的时钟脉冲输入。
- 不支持精度低于一个周期的 PWM 输出脉冲。
- 不支持大于 8 位的比较。
- 在异步计数器模式下，不能使用小于一个周期的时钟输入。
- 不能在 RC0/RC1 引脚上显示振荡器波形。
- 不能模拟串行 I/O。

12.5.3 I/O 时序

在每个指令周期内，软件模拟器仅处理一次外部时序。不能模拟小于一个指令周期的瞬态信号 （如 MCLR 引脚上的尖峰电压）。

注：	在下一指令周期前向软件模拟器注入激励。

注：

第 13 章 MPLAB SIM 入门

13.1 简介

如果您对 MPLAB IDE 和软件模拟器还不熟悉，请参考以下主题以帮助设置 MPLAB IDE，使能够在该环境中使用 MPLAB SIM：

- **第 2 章 “MPLAB IDE 入门：基础教程”**
- **第 3 章 “步骤和教程”**

一旦熟悉了 MPLAB IDE 下的基本软件模拟器操作，就可参见以下主题以帮助使用软件模拟器特有的功能。

- 使用跑表
- 使用激励
- 使用软件模拟器跟踪
- 使用外部存储器

13.2 使用跑表

从 *Debugger>Stopwatch* 打开相应对话框。

指令周期和时间

在程序运行的过程中，软件模拟器会更新 Instruction Cycles 和 Time 域，包括时间单位。

- 单击 Synch （同步）将 stopwatch （跑表）的值与 total simulated （模拟总数）的值同步。
- 单击 Zero （归零）随时将 Instruction Cycles 和 Time 的值设置为零。
- 选择 “Clear Simulation Time On Reset”（在复位时清零模拟时间）以在程序复位时将 Instruction Cycles （指令周期）和 Time （时间）的值设置为零。

注：	在使用激励时应始终选中此选项。

处理器频率

可从 *Debugger>Settings* 对话框的 Clock （时钟）选项卡或 Osc/Trace （振荡器 / 跟踪）选项卡设置此域以及频率的单位。

13.3 使用激励

软件模拟器激励是一个功能强大的工具，使用它可控制到软件模拟器的输入并测试应用程序代码。以下各章详细介绍了激励控制。

- **第 14 章 “使用激励”**
- **第 15 章 “使用激励—— PIC17 器件”**

13.4 使用软件模拟器跟踪

此处将讨论如何使用 MPLAB SIM 软件模拟器跟踪。

- 跟踪执行
- Trace 窗口
- 导出跟踪缓冲器

13.4.1 跟踪执行

跟踪允许记录每一步的代码执行并检验此记录。

注：	在使用跟踪时，模拟速度会减慢。

- 当在 Settings 对话框的 Trace/Pins （跟踪 / 引脚）选项卡或 Osc/Trace （振荡器 / 跟踪）选项卡中启用了跟踪，软件模拟器就会在跟踪缓冲器中记录指令的执行过程。
- 当执行停止时，如通过使用 Halt （暂停）、 Break （中断）或 Breakpoint （断点），跟踪也将停止。
- Trace 窗口在打开的时候会显示跟踪缓冲器的内容。当得到请求时它将更新自身的内容。

13.4.2 Trace 窗口

从 *View>Simulator Trace* 打开 Trace 窗口。

如果跟踪缓冲器为空，该窗口中显示 "No items to display"（没有可显示的项）。否则，该窗口会显示跟踪缓冲器的内容。在执行指令时对指令的跟踪会被写入一个 8192 行的缓冲器。当此缓冲器满时，下一次跟踪将重新写入缓冲器的顶部。

欲知更多 Trace 窗口及其上下文菜单的信息，请参阅**第 8 章 "MPLAB IDE 窗口"**。

13.4.3 导出跟踪缓冲器

通过在 Trace 窗口的鼠标右键菜单中选择 Export 就会显示 Export As 对话框。

- File name——为将要写入跟踪数据的文本文件命名
- Save as type——已经选择为跟踪文件的文件类型 （*.trc）
- Start Cycle、 End Cycle （开始周期、结束周期）——指定要写入的周期的范围

13.5 使用外部存储器

某些 Microchip 器件允许使用外部 （片外）存储器来扩展或替换程序存储器资源。此处将讨论在外部 （片外）存储模式下，软件模拟器如何工作。

- 访问外部存储器
- 模拟外部存储器访问

13.5.1 访问外部存储器

此处将讨论可由选定器件访问的外部存储器容量以及访问的方法。

13.5.1.1 PIC17CXXX 程序存储器模式

PIC17CXXX 器件可以结合使用片上和外部程序存储器，工作在几种程序存储器模式之一。可用的程序存储器模式如下：

- **微处理器**模式仅允许访问外部程序存储器；而忽略片上程序存储器的内容。程序计数器允许访问线性程序存储空间并定义可访问的程序存储器容量 （参见**第 13.5.1.4 节 "程序计数器"**）。微处理器模式是未编程器件的默认模式。
- **单片机**和**受保护的单片机**模式仅允许访问片上程序存储器。尝试读取片上存储器物理限制范围之外的存储单元将会获得未知数据。受保护的单片机模式还启用了代码保护功能。
- **扩展的单片机**模式允许将内部和外部程序存储器作为一个区块访问。在这种模式下，此类器件可以访问整个片上程序存储器；除此之外，还可以访问在程序计数器限定范围之内的外部程序存储器 （参见**第 13.5.1.4 节 "程序计数器"**）。根据需要，程序执行将自动在两个存储器之间切换。

在所有模式下，此类器件均可完全访问数据 RAM。欲知更多信息，请查询器件数据手册的 "存储器构成" 部分。

13.5.1.2 PIC18F6XXX/8XXX 程序存储器模式

某些 PIC18F6XXX/8XXX 器件可以结合使用片上和外部程序存储器，工作在几种程序存储器模式之一。请查阅此类器件的数据手册以确定它是否支持外部存储器。

对于所支持的器件，可用的程序存储器模式如下：

- **微处理器**模式仅允许访问外部程序存储器；而忽略片上程序存储器的内容。程序计数器允许访问线性程序存储器空间并定义可访问的程序存储器容量 （参见**第 13.5.1.4 节 "程序计数器"**）。
- **带引导区的微处理器**模式允许访问从地址 000000h 到引导区末尾的片上闪存存储区。除此之外，还可始终访问在程序计数器限定范围之内的外部程序存储器 （参见**第 13.5.1.4 节 "程序计数器"**）。根据需要，程序执行将自动在两个存储器之间切换。
- **单片机**模式仅允许访问片上闪存存储器。尝试读取片上闪存存储器之外的存储区将会读到全 "0"。（一条 NOP 指令）
- **扩展的单片机**模式允许将内部和外部程序存储器作为一个区块访问。在这种模式下，此类器件可以访问整个片上闪存存储器；除此之外，还可以访问在程序计数器限定范围之内的外部程序存储器 （参见**第 13.5.1.4 节 "程序计数器"**）。有了引导区模式，程序执行可以根据需要，自动在两个存储器之间切换。

在所有模式下，此类器件均可完全访问数据 RAM 和 EEPROM。欲知更多信息，请查询器件数据手册的 "存储器构成" 部分。

13.5.1.3　不带 ROM 的 PIC18C601/801 器件

对于 PIC18C601/801 器件，所有程序存储器地址空间都是外部的。片上程序计数器允许访问线性程序存储空间并定义可访问的程序存储器容量 （参见**第 13.5.1.4 节 “程序计数器”**）。

可以将器件中通用用户 RAM 的最后 512 个字节配置为程序存储器，这种程序存储器被称为 “引导 RAM”。欲知更多信息请参阅器件数据手册。

13.5.1.4　程序计数器

程序计数器的大小将决定可以访问多大容量的程序存储器，例如， 21 位的程序计数器允许访问 2M 字节 （1M 字）的程序存储空间 （片上、片外或两种程序存储器类型的组合）。

13.5.1.5　配置位

对于 PIC17CXXX 和 PIC18F6XXX/8XXX 器件，程序存储器模式可通过使用配置位设置。根据不同的器件，处理器模式选择位位于在对器件编程时编程的配置寄存器中。

对于 PIC18C601/801 器件，外部总线数据宽度被设置为配置位。

欲知更多信息，请查询器件数据手册的 “CPU 的特殊功能”部分。

13.5.1.6　外部存储器接口

外部存储器接口是一种特殊的功能部件，它允许单片机如同访问程序和数据存储器一样访问外部存储器 （如闪存、 EPROM 和 SRAM 等）。

PIC17 器件

当选择了微处理器和扩展的单片机模式时， PORTC、PORTD 和 PORTE 将被配置为系统总线。 PORTC 和 PORTD 被配置为地址 / 数据复用的总线，而 PORTE<2:0> 则用于控制信号。需要外部元器件才能将地址和数据总线解复用。

欲知更多信息，请参见器件数据手册的存储器构成和外部存储器接口部分。

PIC18 器件

使用 MEMCON 寄存器时，可以进行以下配置：

- 使能和禁止外部总线
- 16 位模式——字写模式、位选择模式或位写模式
- 等待——表读 / 写总线周期等待计数 （0—3 个 T_{CY}）

欲知更多信息，请参见器件数据手册的外部存储器接口部分。

13.5.2 模拟外部存储器访问

模拟使用外部存储器的器件需要进行以下步骤：

1. 设置配置位
2. 设置外部存储器

13.5.2.1 设置配置位

要为选定的器件设置配置位的值，请通过选择 _Configure>Configuration Bits_ 打开 Configuration Bits 窗口。在 Category （类别）列中，找到需要设置的位并单击它们以选择它们的值。

> **注：** 也可以使用 `__config` 在代码中设置配置位。欲知更多信息，请参阅器件数据手册和器件头文件 （`.inc` 或 `.h`）。

13.5.2.2 设置外部存储器

要设置 MPLAB IDE 和 MPLAB SIM 以识别外部存储器，请选择 _Configure>External Memory_。然后选中 “Use External Memory”（使用外部存储器）复选框并输入一个范围。

注：



第 14 章 使用激励

14.1 简介

在模拟的过程中，软件模拟器将要执行的程序可能需要外部的激励。激励可以是端口 I/O 引脚的一次电平变化或是一个到端口 I/O 引脚的脉冲。也可能是特殊功能寄存器或其他数据存储器中值的变化。

此外，激励可能需要在模拟过程中的某一指令周期内或某一时间发生。或者，是在某一条件满足时发生，例如，在模拟的过程中，当程序执行到某一指令地址时发生激励。

激励主要分为两种类型：

- 同步——重复而有规律的高低电平序列，具有可调节的占空比，占空比可由每个高 / 低电平时间的时钟周期数设置。
- 异步——由您，用户触发激励。

要定义程序何时发生外部激励、发生何种外部激励及如何发生，可使用以下对话框：

- SCL Generator 对话框
- Stimulus Controller 对话框

如果将使用多种形式的激励输入，应注意输入间的相互作用 （参见**第 14.4 节 "激励输入相互作用"**）。

14.2 SCL GENERATOR 对话框

在 SCL Generator 对话框中使用强大的后台软件模拟器控制语言 （Simulator Control Language， SCL）来创建同步激励。如果只想使用异步激励，可直接转到 Stimulus Controller 对话框进行设置。

SCL Generator 对话框允许输入激励信息，这些信息保存在 workbook 文件中。要打开新的 workbook 文件，选择 *Debugger>SCL Generator> New Workbook*。要打开并编辑已存在的 workbook 文件，选择 *Debugger> SCL Generator>Open Workbook*。

SCL Generator 对话框具有下面列出的选项卡。所有的选项卡具有相同的默认单位值。

- Pin/Register Actions （引脚 / 寄存器动作）
- Advanced Pin/Register （高级引脚 / 寄存器）
- Clock Stimulus （时钟激励）
- Register Injection （寄存器注入）
- Register Trace （寄存器跟踪）

当设置多个选项卡时，请注意输入间的相互作用 （参见**第 14.4 节 "激励输入相互作用"**）。

默认单位值

SCL 发生器的所有选项卡均采用如下单位制：

- 时间单位均为十进制，即指令周期 （cy）、毫秒 （ms）、微秒 （us）和纳秒 （ns）。
- PC 值均为十六进制。
- 引脚值只能为 0 或 1。
- 寄存器值均为十六进制。
- 位域值均为二进制。

标签结构

标签格式必须为：

alpha[_|alpha|numeric]

其中：

- **alpha** 为字母字符
- **_** 为下划线
- **numeric** 为数字字符

所有标签均有如下限制：

- 必须以字母开头
- 禁止以下划线 （_）结尾
- 禁止包含两个相连的下划线

将 SFR 值用作触发信号

只有当用户而不是外设更新 SFR 时，才会发生触发和跟踪。

例如，触发对话框被设置为当 `TMR2 = 0x06` 时触发激励。但当 TMR2 值递增并超过了 0x06 时，仍不会产生触发信号。然而，若在用户代码中执行了如下语句：

```
MOVLW 0X06
MOVWF TMR2
```

则就会产生触发信号。

14.2.1 Pin/Register Actions

可以在该选项卡上输入基本的同步引脚和 / 或寄存器动作。更复杂的动作请使用 Advanced Pin/Register 选项卡。

在此选项卡上输入数据：

1. 在 “Time Units”（时间单位）列中选择用于触发所有激励的时间单位。
2. 单击“Click Here to Add Signals”（单击此处添加信号）文本打开 Add/Remove Signals （添加 / 移除信号）对话框 （参见**第 14.2.3 节 “Add/Remove Signals 对话框”**）。在此对话框中，选择引脚、寄存器或其他将应用激励的信号。这些选项将成为相应列的标题。
3. 填写每一行，即输入触发时间 （“Time”）和每个引脚 / 寄存器列的值。

注：	当将 SFR 值用作触发信号时，请格外小心。

4. 要删除某行，请选中该行，然后单击 **Delete Row** （删除行）。

一旦填写好该选项卡，就可以继续填写其他选项卡或单击 **Generate SCL From All Tabs** （从所有的选项卡生成 SCL）来创建 workbook 文件了。

在信号间滚动：

- 如果在对话框中添加的信号超过了可显示的窗口区域，将出现滚动条。
- 可以在所有的信号间滚动来查看它们的值。

> **注：** 当在信号列间滚动时， Time 列保持不变。

14.2.2 Advanced Pin/Register

可以在该选项卡上输入高级 （复杂）的同步引脚和 / 或寄存器动作。基本动作请使用 Pin/Register Actions 选项卡。

先定义条件，再定义触发。

14.2.2.1 DEFINE CONDITIONS

在本节表 14-1 所示的每一行中，定义一个或多个激励的产生条件。

表 14-1： 定义激励条件

项	定义
条件	当在任何其他列中输入数据时，会自动生成指定的条件的名称。该名称用来标识选项卡上 Define Triggers （定义触发）部分中 Condition 列内的条件。
When （条件）	定义条件。即，当第 2 列内的引脚 / 寄存器的值 （在第 1 列中指定了其类型）与第 4 列中的值具有在第 3 列中指定的关系时，条件为真。 **第 1 列**：选择引脚 / 寄存器类型，在 “SFR”、“Bitfield” 和 “Pin” 中选择一个，或选择 “All” 以选择上面的全部 3 种类型。类型选择操作会筛选第 2 列中的内容。 **第 2 列**：选择将应用条件的引脚 / 寄存器。 **第 3 列**：选择条件，在等于 (=)、不等于 (!=)、小于等于 (<=)、大于等于 (>=)、小于 (<) 或大于 (>) 中选择一个。 **第 4 列**：输入条件值。 **注意：**当将 SFR 值用作触发信号时，请格外小心。
Wait （等待）	当 When 中定义的条件为真时，用于指定应用激励等待的时间长度。 **第 1 列**：等待时间值。 **第 2 列**：等待时间值的单位。
Comments （备注）	添加有关条件的描述信息。

设置一个这样的条件 COND1，当 PORTC 寄存器的值等于 FF 时， 10 ms 后应用在 “Define Triggers”（定义触发）中定义的激励。

例 14-1： 寄存器等于某一值

Define Conditions

Condition	When				Wait		Comments
COND1	SFR	PORTC	=	FF	10	ms	

14.2.2.2 DEFINE TRIGGERS

参照在 “Define Conditions” 中设置的条件，定义本节表 14-2 所示的每一行中的激励触发。

表 14-2： 定义激励触发

项	定义
Enable （启用）	启用或禁止在该行中设置的触发。
Condition （条件）	参照在该选项卡中的 Define Conditions 部分设置的条件的标签。从列表中选择一条件。
Type （类型）	选择触发条件的应用类型，一次 （1x）或连续 / 重复 （Cont）。
Re-Arm Delay （后续触发延时）	如果 Type = Cont，则需要输入再次检查触发条件前的延时。在第一列中输入延时值，在第二列中选择延时单位。
Click Here to Add Signals （单击此处添加信号）	单击 “（Click here to add/remove signals）” 文本打开 Add/Remove Signals 对话框 （参见第 **14.2.3 节 “Add/Remove Signals 对话框”**）。在此对话框中，选择引脚、寄存器或其他应用激励的信号。这些选择会成为相应列的标题。

在信号间滚动：

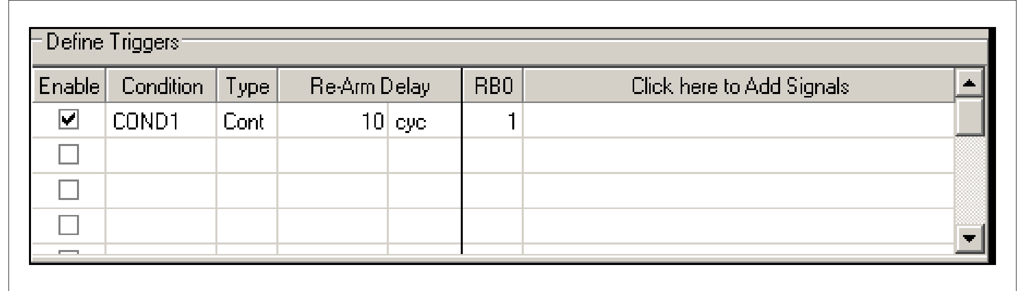
- 如果添加的信号超过了可显示的窗口区域，将出现滚动条。
- 可以在所有信号间滚动来查看它们的值。

> **注：** 当在信号列间滚动时，信号列左边的几列保持不变。

根据上例中设置的条件 COND1，设置如下激励触发。

1. 把 RB0 引脚置为高电平，作为条件 COND1 的第一个实例。
2. 等待 10 个指令周期后再次检查条件。若条件满足，则再次将 RB0 引脚置为高电平。
3. 重复步骤 2 直到程序暂停。

例 14-2： 改变寄存器的值将引脚设置为高电平



14.2.2.3 “DEFINE CONDITIONS” 中的 WAIT 与 “DEFINE TRIGGER” 中 RE-ARM DELAY 的比较

在 “Define Conditions” 中， Wait 时间是指从条件为真到应用激励之间的时间。而在 “Define Trigger” 中， Re-Arm delay 是指从应用激励到再次检查触发条件之间的时间。

14.2.3 Add/Remove Signals 对话框

使用该对话框来添加或移除引脚或寄存器，这些引脚或寄存器将根据相关的 Pin/Register 选项卡上的设置应用激励。

添加信号：

- 从 “Available Signals”（现有信号）下拉框中选择信号类型，“SFR and Field”（SFR 及其位域）或 “Pin Only”（仅引脚）或者 “All Signals”（所有信号）。
- 选用以下一种方法将信号添加到列表中：
  - 选中信号，然后单击 **Add** 把信号添加到 “Selected Signals”（已选信号）列表中。
  - 双击信号，把信号添加到 “Selected Signals” 列表中。

> **注：** 同时支持外设 I/O （如 CCP2）和数字 I/O （如 RB3 或 RC1）的引脚，若配置正确并受软件模拟器支持，仍可使用外设 I/O 功能，但是将仅以数字 I/O 名列出。

该信号将会显示为相关的 Pin/Register （引脚 / 寄存器）选项卡上列的标题。

删除信号：

- 选用以下一种方法将信号从 “Selected Signals” 列表中删除：
  - 选中信号，然后单击 **Remove** 将信号从列表中删除。
  - 双击信号，将信号从列表中删除。

该信号现在将不再显示为相关的 Pin/Register 选项卡上列的标题。

改变 Pin/Register 选项卡上信号的顺序：

- 单击 “Selected Signals” 列表中的信号。
- 单击 **Move Up** 或 **Move Down** 改变该信号在列表中的顺序。

相关的 Pin/Register 选项卡上列标题的顺序将反映该信号的位置。

14.2.4 Clock Stimulus

如表 14-3 所示，应用于引脚的脉冲 （高电平和低电平）的值称为时钟激励。可在此处输入时钟激励。

表 14-3： 定义时钟激励

项	定义
Label （标签）	命名指定的时钟激励 （可选）。
Pin （引脚）	选择将应用时钟激励的引脚。
Initial （初态）	输入时钟激励的初始状态，高电平或低电平。
Low Cycles （低电平周期数）	输入低电平的周期数 （以时钟脉冲计数）。
High Cycles （高电平周期数）	输入高电平的周期数 （以时钟脉冲计数）。
Begin （开始）	单击此处以反映在 Begin 部分中的选择： Always （默认） 程序运行时立即开始激励。 PC= 当程序计数器等于输入的值时开始激励。 Cycle= 当指令周期计数等于输入的值时开始激励。 Pin= 当选定的引脚具有选定的值 （低电平或高电平）时开始激励。
End （结束）	单击此处以反映在 End 部分中的选择： Never （默认） 应用激励直到程序暂停。 PC= 当程序计数器等于输入的值时结束激励。 Cycle= 当指令周期计数等于输入的值时结束激励。 Pin= 当选定的引脚具有选定的值 （高电平或低电平）时结束激励。
Comments （备注）	添加有关激励的描述信息。

14.2.5 Register Injection

如表 14-4 所示，可以把文件中设置的值注入到寄存器。可在此处输入寄存器注入信息。

表 **14-4**： 定义寄存器注入

项	定义
Label （标签）	命名指定的寄存器注入 （可选）。
Destination （目标寄存器）	选择将要注入数据的目标寄存器 *。
Trigger （触发）	选择触发注入的时间，按要求 （Demand）或当 PC 等于一个指定的值时 （参见下一栏）。 若外设可用，那么只能按要求触发。若外设不可用，则必须按 PC 触发。例如，若 ADON 位置 1 并且 ADRESL 寄存器可访问，那么就将发生对 ADRESL 的注入。
PC Value （PC 值）	如果 Trigger=PC，输入一个触发注入所需的 PC 值。
Data Filename （数据文件名）	浏览以找到注入文件。
Rewind （反卷）	**Yes**——将文件中所有的数据注入了寄存器之后，就将回到文件的开头重新开始。 **No**——将文件中所有的数据注入了寄存器之后，将继续使用最后一值注入。
Format （格式）	选择注入文件的格式。 **Hex**——ASCII 十六进制格式 **Raw**——Raw 图像格式，以二进制形式表示。 **SCL**——SCL 格式。参见定义规范。 **Dec**——ASCII 十进制格式
Comments （备注）	添加有关寄存器注入的描述信息。

* 可能不会显示所有的寄存器。例如，选择注入 PIC18F458 中的 ADRESL 寄存器实际上会将激励注入到 ADRESH 和 ADRESL 两个寄存器。

例 **14-3**： 十六进制格式的寄存器激励文件

```
110
02E
A38
541
1A0
0FD
```

14.2.6 Register Trace

如表 14-5 所示，在某些条件下，程序运行过程中可能会将指定寄存器的值保存到文件中 （跟踪）。可在此处指定寄存器跟踪信息。

表 **14-5**： 定义寄存器跟踪

项	定义
Label （标签）	命名指定的寄存器跟踪 （可选）。
Source （源寄存器）	选择要跟踪的源寄存器。
Trigger （触发）	选择触发跟踪的时间，按要求 （Demand）或当 PC 等于一个指定的值时 （参见下一栏）。 若外设可用，那么只能按要求触发。若外设不可用，则必须按 PC 触发。例如，若 ADON 位置 1 并且访问 ADRESL 寄存器，将跟踪 ADRESL。
PC Value （PC 值）	如果 Trigger=PC，输入一个触发跟踪的 PC 值。
Trace Filename （跟踪文件名）	浏览以找到跟踪文件的位置。
Format （格式）	选择跟踪文件的格式。 **Hex**——ASCII 十六进制格式 **Raw**——Raw 图像格式，以二进制形式表示。 **SCL**——SCL 格式。参见定义规范。 **Dec**——ASCII 十进制格式
Comments （备注）	添加有关寄存器跟踪的描述信息。

14.3 STIMULUS CONTROLLER 对话框

使用 Stimulus Control （激励控制）对话框控制 MPLAB SIM 使用激励的方式。

可以在 Stimulus Control （激励控制）对话框中设置激励类型并把此信息保存到情形中。要打开新的情形，选择 *Debugger>Stimulus Controller>New Scenario*。要打开并编辑已存在的情形，选择 *Debugger> Stimulus Controller >Open Scenario*。

当设置 SCL 文件输入和异步激励时，请注意激励输入相互作用。

14.3.1 Stimulus (SCL) File （激励文件）

如果已经用 SCL Generator 对话框创建了同步激励文件，单击 **Attach** （连接）搜索文件及其位置，并将文件与控制器相连接。

要从控制器中移除 SCL 文件，单击 **Detach** （分离）。

如果想把两个 SCL 合并为一个文件，单击 **Import/Merge** （导入 / 合并）。

14.3.2 Asynchronous Stimulus （异步激励）

在此逐行输入任何异步或用户触发激励。要删除行，选中要删除的行，然后单击 **Delete Row** （删除行）。

项	定义
Fire （触发）	单击此列中与欲触发激励的行所对应的按钮。显然，在能使用 Fire 前，必须对行中其他所有的项进行设置。
Pin （引脚）	选择或改变将应用激励的引脚。在下拉列表中列出了可用的引脚名。实际引脚名取决于在 IDE 中选择的器件。
Action （动作）	选择或改变激励动作。可以选择 Pulse （脉冲）、 High （高电平）、Low （低电平）或 Toggle （翻转）其中之一。
Width （宽度）	如果 Action （动作）选择为 Pulse （脉冲），那么可以在此指定或改变脉冲宽度。在相应行的后一单元中输入该值的单位。
Units （单位）	如果 Action 选择为 Pulse，那么可以在此指定或改变脉冲宽度单位。在相应行的前一单元中输入该单位的值。
Comments （备注）	指定或改变有关激励的备注。

14.4 激励输入相互作用

如果引脚 （如 RB1）分配和端口 （如 PORTB.RB）分配发生在同一个周期，端口分配会取代引脚分配。但是，如果它们发生在不同的周期，则可以混合并匹配引脚分配和端口分配。

此外，还可以通过引脚 / 端口分配来使用 PORT （端口）注入。

注：



第 15 章 使用激励—— PIC17 器件

15.1 简介

激励功能允许在调试代码时使软件模拟器应用人工激励。可以将引脚设置为高电平或低电平或者直接向寄存器中注入值。

在 Debugger 菜单中选择 Stimulus，然后选择激励类型选项卡。

本章涉及下列主题：

- 使用引脚激励——I/O 引脚上的同步或异步激励
- 使用文件激励——I/O 引脚或文件寄存器上的触发激励，触发激励是在文件中设置的

15.2 使用引脚激励

引脚激励由用户定义。它可能是：

- 异步——由用户触发。
- 同步——有规律重复的占空比可调的高 / 低电平序列，占空比由高 / 低电平时间中的时钟周期数决定。

可在程序计数器地址将引脚设置为高电平或低电平。在模拟过程中将会依次查看这些地址。

可以编辑、保存或从文件恢复引脚激励列表中的内容。

注：	建议不要在 Stimulus 对话框外修改激励文件。

- 创建 / 编辑引脚激励
- 应用引脚激励
- 引脚激励显示

15.2.1 创建 / 编辑引脚激励

1. 选择 *Debugger>Stimulus Controller* 然后单击 **Pin Stimulus** 选项卡。
2. 要创建新文件，单击 **Add Row** （添加行）以创建一新的数据输入行。要打开一个现有的引脚激励文件进行编辑，单击 **Load** （加载）在 “打开”对话框中选择文件。
3. 单击要添加或更改数据的表格行的 “Type”（类型）列。选择或更改激励类型（Async （异步）或 Synch （同步））。根据所选的激励类型，每个单元格将得到相应的控制以允许您指定该单元格的值。

异步设置

a) 单击 “Pin”（引脚），选择或更改将应用激励的引脚。其中列出了可用的引脚名称。实际名称取决于在 IDE 中选定的器件。
b) 单击 “Action”，选择或更改激励动作。可选择 Pulse、 High、 Low 或 Toggle。
c) 单击 “Comments”（备注），输入或更改备注，备注将保存在引脚激励文件中并可从中恢复。

同步设置

a) 单击 “Pin”，选择或更改将应用激励的引脚。其中列出了可用的引脚名称。实际名称取决于在 IDE 中选定的器件。
b) 单击 “High Cycles”（高电平周期数），设置或更改在应用高电平状态前的时钟周期数。单击 “Low Cycles”（低电平周期数），设置或更改在应用低电平状态前的时钟周期数。
c) 单击 “Invert”（反相），设置或更改是否将高低电平反相。
d) 单击 “Comments”，输入或更改备注，备注将保存在引脚激励文件中并可从中恢复。

4. 当添加完或编辑完行中数据后，单击 **Edit Complete** （完成编辑）。
5. 继续添加或编辑其他数据行直到完成引脚激励设置。若想要移除某行，请单击该行中的某个单元格以选中此行，然后单击 **Delete Row**。
6. 完成后单击 **Save** 保存引脚激励文件 （.psti）。

15.2.2 应用引脚激励

一旦创建或编辑完引脚激励文件后，就可按以下方法应用激励：

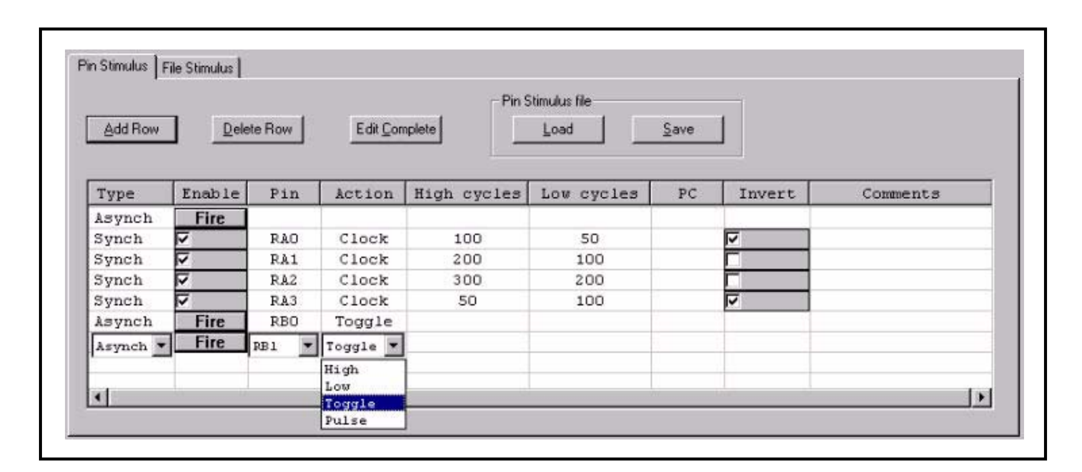
- 异步激励 （Type = Asynch）：按 Fire 按钮。在指定的引脚上产生指定的事件。
- 同步激励 （Type = Synch）：选中 Enable 域。向 MPLAB SIM 发送一条消息以更新激励。这样，所显示的激励均是最新的。

注：	要激活激励，必须打开 Stimulus 对话框。

15.2.3 引脚激励显示

选择 *Debugger>Stimulus*，然后单击 Pin Stimulus 选项卡以设置引脚激励。关闭 Stimulus 对话框会禁止激励。

> 注 **1：** 若在外部更改了任何文件，当通过该对话框的选项卡使用该文件时，这些更改将不会反映到要传递给软件模拟器的激励中。建议不要在 Stimulus 对话框外更改激励文件。
>
> **2：** 当关闭该对话框时，将丢失未保存的更改，而不发出通知。



按钮——行编辑

- Add Row——允许添加新的引脚激励。将会在引脚列表底部创建一默认引脚行，光标将移至此处以进行编辑。
- Delete Row——删除列表中光标所在的行。
- Edit Complete——关闭任何列表框的显示，因此将清晰地显示所有的数据。

按钮—— 引脚激励文件

- Load——允许指定先前保存过的引脚激励文件。并将其内容添加到激励列表中。
- Save——允许将激励列表中的当前内容保存到您选择的引脚激励文件中。

表格项

- Type——具有包含 “Asynch” 和 “Synch” 的选择列表。所选的类型决定该行中其他单元格内哪些项可用。
- Enable——具有一个 Synch 激励复选框和一个 Asynch 激励的 Fire 按钮。 Enable 不是一个数据单元格。
- Pin——包含可用引脚的名称。实际的名称取决于在 IDE 中选定的器件。

- Action

  Type = Asynch：

  Pulse：将引脚电平状态翻转并立即返回。

  High：将引脚电平状态改为高电平。

  Low：将引脚电平状态改为低电平。

  Toggle：将引脚电平状态翻转。

  Type= Synch：Clock
- High cycles——应用于同步激励的高电平周期数。
- Low cycles——应用于同步激励的低电平周期数。
- Invert——当在同步激励模式下，选中该复选框将使高低电平反相。
- Comments——若要将引脚激励保存在一个文件里，那么可指定一个备注，备注可保存入该文件并可从该文件中恢复。

15.3 使用文件激励

MPLAB SIM 触发通过使用文件激励，将引脚和寄存器的值改变为在激励文件中指定的值。这些文件指定当触发启动时将发送到引脚或寄存器的值。触发可能是对引脚电平的周期计数或者是寄存器的程序存储器地址。

> 注 **1：** 建议不要在 Stimulus 对话框外更改激励文件。
>
> **2：** 如果在端口引脚上已设置了引脚激励，那么将不能用文件激励向该引脚或相应的寄存器注入激励。

- 创建 / 编辑文件激励
- 应用文件激励
- 文件激励显示

15.3.1 创建 / 编辑文件激励

选择 *Debugger>Stimulus Controller* 然后单击 **File Stimulus** 选项卡。文件激励文件由一个或多个同步激励文件组成。同步激励文件包含关于触发的信息，这些触发用于向引脚或寄存器应用激励。寄存器激励在寄存器激励文件中指定。

```
MyStimulus.fsti
    SyncStimulus1.ssti
    SyncStimulus2.ssti
        TRISBStimulus.rsti
    SyncStimulus3.ssti
        TRISBStimulus.rsti
        T1CONStimulus.rsti
```

- 创建 / 编辑文件激励文件 （.fsti）
- 创建 / 编辑同步激励文件 （.ssti）
- 创建 / 编辑寄存器激励文件 （.rsti）

15.3.1.1 创建 / 编辑文件激励文件 （.FSTI）

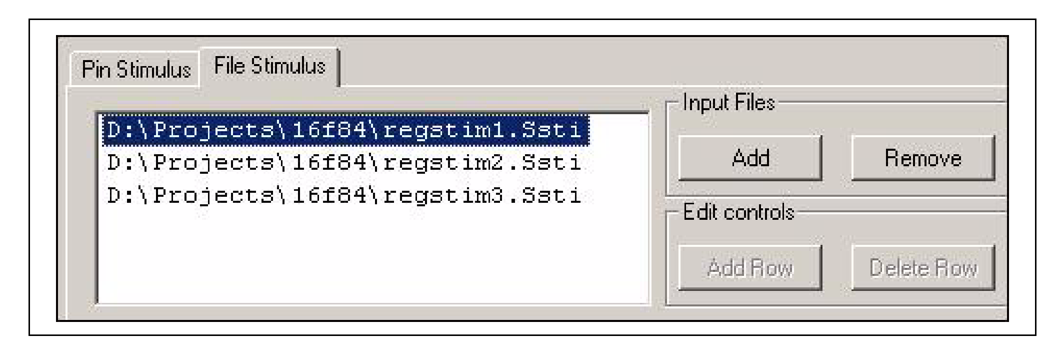
文件激励文件包含一个或多个将应用于软件模拟器的同步激励文件 （*.ssti）。

要创建文件激励文件：

1. 在文件列表中 （选项卡下）创建或打开一个或多个同步文件。
2. 在 File Stimulus 下单击 **Save Setup** （保存设置）。

要编辑一个现有的文件激励文件：

1. 在 File Stimulus 下单击 **Load Setup** （加载设置），打开现有的文件激励文件。
2. 使用文件列表和 "Input Files"（输入文件）下的 **Add** 和 **Remove** 按钮编辑列表。



15.3.1.2 创建 / 编辑同步激励文件 （.SSTI）

1. 如果在文件列表中 （选项卡下）没有列出激励文件，单击 "Input Files" 下的 **Add**，打开 "打开" 对话框。浏览到要创建新文件的位置或要编辑的现有文件的位置。要创建新的激励文件，请输入文件名并单击 **Open**。要打开现有的文件进行编辑，单击选中该文件然后单击 **Open**。现在激励文件的位置和名称应显示在文件列表中。
2. 单击文件列表中的激励文件，以选中该文件进行编辑。然后单击 **Edit**。若发现选错了文件，单击 **Cancel** 然后选择另一个文件。
3. 对于新的文件，单击 **Add Row** 添加一条新的数据输入行。
4. 在要添加或更改数据的表格行里单击 "Trigger On"（启动触发）列。选择或更改触发类型 （Cycles 和 PC）。根据所选的触发类型，每个单元格将得到相应的控制以允许指定单元格的值。

   周期设置

   a) 单击 "Trig Value"（触发值），设置启动触发的周期数。
   b) 从 "Pin/Register" 列中选择引脚，当启动触发时该引脚将应用下一列中的值。实际可用的引脚名称取决于选定的设备。

注：	不要选择寄存器。寄存器不会应用周期激励。

   c) 单击 "Value" 输入或更改当发生触发时要应用的值。对于 Pin，该值只能为 0 或 1。将会依次应用这些值，每个触发事件应用一个值。
   d) 要设置其他引脚，单击 **Add Column** （添加列）以添加 "Pin/Registe" 和 "Value" 列。
   e) 单击 "Comments"，输入或更改文本串，该备注将保存在文件中并可从中恢复。

PC 设置

a) 单击 “Trig Value”，设置启动触发的 PC 地址。

b) 从 “Pin/Register”中选择寄存器，当启动触发时该寄存器将应用下一列中的值。实际可用的寄存器名称取决于选定的器件。

> **注：** 不要选择引脚。引脚不能应用 PC 激励。

> **注：** 不能向端口寄存器 （如 PORTB）注入文件值。对这种寄存器只能使用引脚值。同样，也不能向 W 寄存器注入文件值。

c) 单击 “Value”输入或更改当发生触发时要应用的值。对于寄存器，则需输入激励文件 （`*.rsti`）的名称。该文件包含一系列值，这些值将会被依次应用，并且一个触发事件仅应用一个值。

d) 单击 “Comments”，输入或更改文本串，该备注将保存在文件中并可从中恢复。

e) 对于寄存器激励，一个激励文件可能只能与一个寄存器关联，反之亦然。如果添加了其他的行 / 列，只有最后添加的行 / 列才会被执行。

f) 如果想要删除行，单击行中的单元格以选中该行，然后单击 **Delete Row**。

5. 完成后单击 **Save** 保存同步激励文件。

15.3.1.3 创建 / 编辑寄存器激励文件 （.RSTI）

寄存器激励文件包含寄存器激励信息，这些信息是十六进制格式的，并以回车符分隔。任何文本编辑器或电子表格应用程序都可用来创建或编辑该文件。默认扩展名为 .rsti，但也可使用任何其他扩展名。在同步激励文件 （*.ssti）的 Value 列中将寄存器激励文件指派给一个寄存器。

> **注：** 一次只可把一个寄存器激励文件指派给一个寄存器。

例 15-1： 寄存器激励文件

```
10
2E
38
41
A0
FD
```

15.3.2 应用文件激励

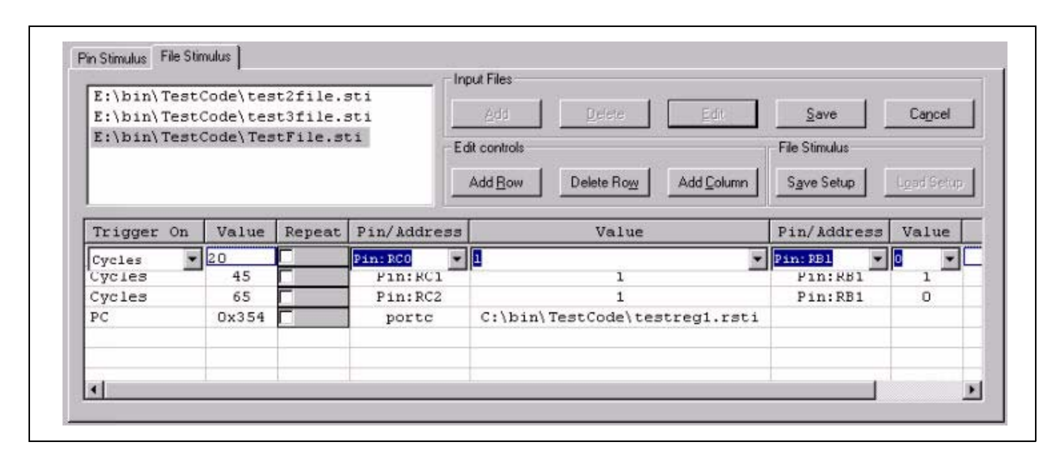
一旦创建 / 编辑了文件激励文件，当在 MPLAB IDE 中运行程序时就会应用该激励。当指定的触发激活时，就会发生指定的事件。

> **注：** 要激活激励， Stimulus 对话框必须保持打开。

15.3.3 文件激励显示

选择 *Debugger>Stimulus Controller*，然后单击 File Stimulus 选项卡以设置文件激励。关闭 Stimulus 对话框会禁止激励。

注	
1：	只有选择了 MPLAB SIM 工具，在 Debugger 菜单中才会出现 Stimulus Controller （激励控制器）菜单。
2：	若在外部更改了任何文件，当通过该对话框的选项卡使用该文件时，这些更改将不会反映到要传递给软件模拟器的激励中。建议不要在 Stimulus 对话框外更改激励文件。
3：	当关闭该对话框时，将丢失未保存的更改，而不发出通知。



文件列表

可用作文件激励的输入文件列表。使用 Input Files 下的各个按钮可添加、删除、编辑和保存该列表中的文件。

按钮—— **Input Files**

- Add——打开一个浏览对话框从中选择一个文件以将其添加到文件列表中。可以选择一个现有的文件或是输入要创建的文件的名称。
- Delete——将选定的文件从列表中删除 （而不是从驱动器中删除）。
- Edit——显示在激励列表中选中的文件的内容，并允许编辑该内容。该操作还将禁用 Add 和 Delete 按钮。
- Save——将激励列表中的内容保存到一个文件中，同时清空该激励列表，禁止编辑并重新使能 Add 和 Delete 按钮。
- Cancel——取消文件选择。

注：	一次只能显示和编辑文件列表中的一个文件。

按钮——**Edit Controls** （编辑控制）

- Add Row——在激励列表的底部创建一默认行并将光标移到此处以进行编辑。
- Delete Row——将光标所在处的行从列表中删除。
- Add Column——将另一对引脚 / 寄存器值列添加到激励列表中 Comments 列的左边，以允许为每个触发指定其他激励。

按钮—— File Stimulus

这些按钮使您可以保存和恢复组合的文件设置。

- Save Setup 将目前存在于文件列表中的文件的名称保存到一个文件中。
- Load Setup 将文件中的文件名加载到文件列表中，就如同将文件逐一添加到列表中一样。

注：	目前还没有方法查看 File Stimulus 设置的组合内容。

表格项

- Trigger On——在 Cycles 和 PC 间选择。所选择的类型决定该行其余的一些单元格内哪些项可用。
- Trig Value——启动触发的周期数或 PC 地址。输入的值可以为十进制或十六进制数据。十六进制数必须以 0x 开头。
- Pin/Register——应用值的对象；可从可用的引脚和寄存器的名称中选择。实际名称取决于在 IDE 中选定的器件。
- Value——当触发发生时，将应用的值。对于 Pin，该值只能为 0 或 1。对于 Register，为一个值或包含一系列值的文件名。这些值将会被依次应用，并且一个触发事件仅应用一个值。
- Comments——可保存到文件并可从中恢复的一串文本。

注：	只有使用 PC 触发时 Register 激励才会有效。可以给 Register 指定 Cycle 触发，但这是无效的。此外， Register 值必须从文件指定；指定一个数字值将无效。

第 16 章 软件模拟器疑难解答

16.1 简介

本章旨在帮助您解决在 MPLAB SIM 使用过程中所遇到的任何问题、错误或难点。若本章中没有您所需的帮助信息，请参见前言中的联系方式联系 Microchip Technology。

- 常见问题 /FAQ
- 限制

16.2 常见问题 /FAQ

软件模拟器不能使用我创建的激励文件

即使文件不正确，软件模拟器也不会通知您；而是直接忽略文件的内容。仅应使用以下一种方法来设置激励文件。

- 使用 SCL Generator 和 Stimulus Control 对话框
- 使用 Pin Stimulus 和 File Stimulus 对话框 （PIC17 器件）

不要在外部编辑激励文件。

此外，可能还要考虑引脚 / 端口间的相互作用。请参阅与正在使用的激励类型相对应的文档。

我不能将多个寄存器激励文件关联到一个寄存器

对于 PIC17 MCU 器件，在激励中， MPLAB SIM 只允许将一个寄存器激励文件（.rsti）与一个寄存器相关联。

我不能设置条件断点。

目前还未实现条件断点。

我在 Breakpoint 对话框中设置的符号断点不起作用。

Breakpoint 对话框不能识别符号地址。但是，如果您输入的符号名的开始字符可以表示为一个有效的 16 进制地址，那么断点将会被设置在该地址上。

虽然在代码窗口中似乎设置并使能了断点，但软件模拟器并不识别这些断点。

请确认在 Settings 对话框中的 **Break Options** 选项卡上设置了 “Global Break Enable”（使能全局断点）。

我不能使用 MCLR 引脚复位程序。

对于 dsPIC 或 PIC18F 器件，不能使用引脚来复位器件。必须使用 Debugger 菜单中的复位命令。

我的程序不停地复位

请检查选定器件的配置位设置 （*Configure>Configuration Bits*）。一些复位功能，如 Watchdog Timer Reset 默认情况下是使能的。

看门狗定时器超时时软件模拟器不暂停。

请确认在 Settings 对话框中的 **Break Options** 选项卡上已设置了 “Global Break Enable”并在 Configuration Bits 对话框中已使能了看门狗定时器超时。

16.3 限制

可以在 MPLAB SIM 的在线帮助中找到关于仿真器的一般限制和器件特定的限制。



第 17 章 软件模拟器使用参考

17.1 简介

一旦选择 MPLAB SIM 作为 MPLAB IDE 中的调试工具 （*Debugger> Select Tool>MPLAB SIM*），就可使用以下软件模拟器特定的功能：

- 第 **17.2** 节 **“调试功能”**
- Settings 对话框
- 第 **17.4** 节 **“Settings** 对话框**——PIC17** 器件**”**

17.2 调试功能

会在以下 MPLAB IDE 功能组件中添加软件模拟器特定的调试项：

- Debugger 菜单——标准调试操作外加 Stopwatch 和 Stimulus
- View 菜单——Simulator Trace （软件模拟器跟踪）
- 鼠标右键菜单——标准调试操作
- 工具栏——标准的调试工具栏
- 状态栏——将在此指出使用的调试工具为 MPLAB SIM

17.2.1 Debugger 菜单

除了标准的 MPLAB IDE 调试菜单项之外，还会出现以下工具特定的项：

Stopwatch

用跑表对代码的执行进行计时。欲知更多信息，请参见 **13.2 “使用跑表”**。

Stimulus Controller

对于 PIC17 器件：在代码中设置 I/O 引脚和文件寄存器激励并在执行程序的时候应用它们。欲知更多信息，请参见**第 15 章 “使用激励—— PIC17 器件”**。

对于所有其他器件：使用 Stimulus Control 对话框设置激励。欲知更多信息，请参见**第 14 章 “使用激励”**。

SCL Generator

在 SCL 中开发代码以控制激励触发。欲知更多信息，请参见**第 14 章 “使用激励”**。

Profile （概要文件）

在 Output 窗口中显示软件模拟器当前状态的概要或清零 （复位）当前状态。欲知更多信息，请参见**第 14 章 “使用激励”**。

Refresh PM （刷新 PM）

刷新程序存储器以便重新启动程序。

17.2.2 View 菜单

Simulator Trace

显示一个窗口，该窗口包含在程序执行过程中对当前存储器进行的跟踪。欲知更多信息，请参见**第 13 章 “MPLAB SIM 入门”**。

17.3 SETTINGS 对话框

选择 *Debugger>Settings* 以打开 Settings 对话框。

> **注：** 只有从 Debugger 菜单中选择了软件模拟器，该选项才可用。

此对话框由几个用于设置调试器的特性和功能的选项卡组成。

- **第 17.3.1 节 “Osc/Trace 选项卡”**
- **第 17.3.2 节 “Break Options 选项卡”**
- **第 17.3.3 节 “SCL Options （SCL 选项）选项卡”**
- **第 17.3.4 节 “UART1 I/O 选项卡”**
- **第 17.3.5 节 “Limitations 选项卡”**

17.3.1 Osc/Trace 选项卡

选择 *Debugger>Settings* 然后单击 Osc / Trace 选项卡。

Processor Frequency （处理器频率）

可以在此对话框中将时钟速度设置为希望代码运行的速度。也可以使用这些值在 Stopwatch 中设置一些特定的字段。

1. 选择频率的单位
2. 输入频率的值

> **注：** 配置位的设置对处理器速度没有影响，例如，如果使用 4 倍频的 PLL，软件模拟器仍会以处理器频率运行，而不是 4 倍的处理器频率。

跟踪选项

通过选中 / 清除 “Enable Trace”（使能跟踪）复选框来启用 / 禁止跟踪。如果要在跟踪缓冲器满时停止程序执行，请选中 “Break on Trace Buffer Full”（在跟踪缓冲器满时中断）。

要查看 Trace 窗口，请选择 *View>Simulator Trace*。欲知更多有关跟踪的信息，请参阅**第 13.4 节 “使用软件模拟器跟踪”**。

17.3.2 Break Options 选项卡

选择 *Debugger>Settings* 然后单击 **Break Options** 选项卡。为具体的处理器部分或条件设置中断选项。

产生的消息显示在 Output 窗口中的 MPLAB SIM 标签下。这种消息使用以下标准格式显示：

组件－消息编号：消息

例如：

```
CORE-E0001: Stack Error occurred
ADC-W0001: Cannot change ADCON1 SSRC when ADON=1
```

Core （内核）

- Break （中断）——在发生内核错误时报告错误并中断程序执行
- Ignore （忽略）——忽略内核错误 （产生陷阱，常规操作）
- Report （报告）——在发生内核错误时报告错误并产生陷阱

Peripheral （外设）

- Break （中断）——在发生外设错误时报告错误并中断程序执行
- Ignore （忽略）——忽略外设错误 （产生陷阱 / 复位，常规操作）
- Report （报告）——在发生外设错误时报告错误并产生陷阱

Stack Full / Underflow （堆栈满 / 下溢）

注：	如果已将内核设置为在错误时中断，即使没有将堆栈设置为中断，可能还是要在发生堆栈错误时中断程序执行。这是因为堆栈也是内核的一部分。

- Break （中断）——在发生堆栈错误时中断
- Reset （复位）——在发生堆栈错误时复位 （产生陷阱，常规操作）

WDT Time-out （WDT 超时）

- Break （中断）——在发生 WDT 超时时中断
- Break+Warn （中断＋警告）——在 WDT 超时时报告 WDT 超时并中断程序执行
- Reset （复位）——在发生 WDT 超时时复位 （常规操作）

17.3.3 SCL Options （SCL 选项）选项卡

选择 _Debugger>Settings_ 然后单击 SCL Options 选项卡。设置产生 SCL 错误时的屏幕显示选项。

产生的消息显示在 Output 窗口中的 MPLAB SIM 选项卡下。这种消息使用以下标准格式显示：

组件一消息编号：消息

例如：

```
CORE-E0001: Stack Error occurred
ADC-W0001: Cannot change ADCON1 SSRC when ADON=1
```

SCL

- 在发生 SCL 错误时报告错误并中断程序执行
- 忽略 SCL 错误 （产生陷阱 / 复位，常规操作）
- 在发生 SCL 错误时报告错误并产生陷阱。

17.3.4 UART1 I/O 选项卡

选择 *Debugger>Settings* 然后单击 UART1 I/O 选项卡。

融合了 UART，器件就具备了使用标准 C 函数库 I/O 的功能。当选中“Enable UART1 I/O”（启用 UART1 I/O）选项时，就可使用 UART1 从某个文件中读取数据并将数据写到另一个文件或输出窗口中。

Input

可以通过单击 **Browse** 按钮并使用 “打开”对话框选择要输入的激励文件。

选择 “Rewind Input”（反卷输入）意味着当读到激励文件末尾时，它会返回到文件开始并重新读取，即将会连续循环读取输入的数据。不选中此项意味着当读到激励文件末尾时，将停止读取并不再注入任何激励 （直到复位）。

Output

若要输出文件，选中 “File”并通过单击 **Browse** 按钮，使用 “另存为”对话框选择文件名。

若要将 Output 窗口用作输出，单击 “Window”。

17.3.5 Limitations 选项卡

选择 *Debugger>Settings* 然后单击 Limitation 选项卡。

此对话框显示当前选定器件 （*Configure>Select Device*）的已知 MPLAB SIM 限制。

欲知其他有关器件限制的信息，请单击 **Details**。

17.4 SETTINGS 对话框——PIC17 器件

选择 *Debugger>Settings* 以打开 Settings 对话框。

> 注： 只有从 Debugger 菜单中选择了软件模拟器，该选项才可用。

此对话框由几个用于设置调试器的特性和功能的选项卡组成。

- Clock 选项卡
- Break Options 选项卡
- Trace/Pins 选项卡
- Limitations 选项卡

17.4.1 Clock 选项卡

选择 *Debugger>Settings* 然后单击 Clock 选项卡。

可以在此对话框中将时钟速度设置为希望代码运行的速度。也将使用这些值设置 Stopwatch 的一些特定域。

1. 选择频率的单位
2. 输入频率的值

17.4.2 Break Options 选项卡

选择 *Debugger>Settings* 然后单击 Break Options 选项卡。

Global Break Enable

若选中此选项，将使能程序断点。

Stack 选项

此选项控制堆栈满 / 下溢时的软件模拟器操作。

- Disable Stack Full/Underflow Warning （禁止堆栈满 / 下溢警告）

若选中此选项，软件模拟器在检测到堆栈满和堆栈下溢条件时将不产生警告。

选择以下一种选项以确定当软件模拟器检测到堆栈满和堆栈下溢条件时的操作。

- Break on Stack Full/Underflow （在堆栈满 / 下溢时中断）
- Reset on Stack Full/Underflow （在堆栈满 / 下溢时复位）

注：	Stack 组是禁止的，除非在 Configuration Bits 对话框中启用了 STACK OVERFLOW RESET （堆栈溢出复位）。

WDT 选项

选择以下一种选项以确定当软件模拟器检测到 WDT 超时时的操作。

- Break On WDT time-out （在 WDT 超时时中断）
- Reset on WDT time-out （在 WDT 超时时复位）

注：	如果在 Configuration Bits 对话框中禁止了 WDT， WDT 组也将被禁止。

17.4.3 Trace/Pins 选项卡

选择 *Debugger>Settings* 然后单击 Trace/Pins 选项卡。

MCLR 上拉使能

如果选择此选项， MCLR 上拉将会被使能。

Trace 选项

- Trace Enabled

若选中此选项，将会跟踪指令执行。

- Break on Trace Buffer Full

若选中此选项，软件模拟器将在跟踪缓冲器满时暂停程序执行。

17.4.4 Limitations 选项卡

选择 *Debugger>Settings* 然后单击 Limitation 选项卡。

此对话框显示已知的对当前选定器件 （*Configure>Select Device*）的 MPLAB SIM 限制。

欲知其他有关器件限制的信息，请单击 **Details** 按钮。

注：



词汇表

ANSI

美国国家标准学会 （American National Standards Institute， ANSI）是在美国负责制定和核准标准的组织。

ASCII

美国信息交换标准码是使用 7 位二进制数来表示每个字符的字符集编码。它包括大小写字母、数字、符号和控制字符。

Build

为应用程序编译和链接所有源文件。

C

一种通用的编程语言，具有精简的表达式、现代的控制流和数据结构，以及丰富的运算符集。

COFF

公共目标文件格式。此格式的目标文件包含机器码、调试信息和其他信息。

DSC

*参见*数字信号控制器 （Digital Signal Controller）。

EEPROM

电可擦除可编程只读存储器。一种可电擦除的特殊类型的 PROM。一次可写入或擦除一个字节的数据。即使关断电源， EEPROM 仍能保留自身内容。

EPROM

可擦除可编程只读存储器。通常可通过曝露在紫外线下而擦除的可编程只读存储器。

FNOP

强制性空操作。强制性 NOP 周期是双周期指令的第二个周期。由于 PICmicro 单片机采用流水线架构，所以当它执行当前指令时，会预取物理地址空间内的下一条指令。然而，如果当前指令更改了程序计数器，这条预取的指令将被忽略，引起一个强制性的 NOP 周期。

GPR

通用寄存器。器件数据存储器 （RAM）的一部分，用于常规用途。

Hex 代码 （Hex Code）

以十六进制格式存储的可执行指令代码。 Hex 代码包含在 hex 文件中。

Hex 文件 （Hex File）

一种包含用于对器件进行编程的十六进制地址和值 （hex 代码）的 ASCII 文件。

ICD

在线调试器。 MPLAB ICD 2 是 Microchip 的在线调试器。

ICE

在线仿真器。 MPLAB ICE 2000 和 MPLAB ICE 4000 是 Microchip 的在线仿真器。

IDE

集成开发环境。 MPLAB IDE 是 Microchip 的集成开发环境。

IRQ

*参见*中断请求 （Interrupt Request）。

ISO

*参见*国际标准化组织 （International Organization for Standardization）。

ISR

*参见*中断服务程序 （Interrupt Service Routine）。

Make 项目 （Make Project）

重新编译应用程序的命令，只重新编译那些自从上次编译完成后已更改的源文件。

MCU

单片机器件。 microcontroller 的缩写。另一种缩写是 uC。

MPASM 汇编器 （MPASM Assembler）

Microchip PICmicro 单片机、 KEELOQ 器件和 Microchip 存储器的可重定位的宏汇编器。

MPLAB ASM30

Microchip 供 dsPIC30F 数字信号控制器使用的可重新定位的宏汇编器。

MPLAB C1X

指 Microchip 的 MPLAB C17 和 MPLAB C18 C 编译器。MPLAB C17 是 PIC17 器件的 C 编译器， MPLAB C18 是 PIC18 器件的 C 编译器。

MPLAB C30

Microchip dsPIC30F 数字信号控制器的 C 编译器。

MPLAB ICD 2

与 MPLAB IDE 配合工作的 Microchip 在线调试器。 ICD 支持内置调试电路的闪存器件。每个 ICD 的主要组件均为调试模块。一个完整的系统由模块、适配头、演示板、电缆和 MPLAB IDE 软件组成。

MPLAB ICE 2000/MPLAB ICE 4000

与 MPLAB IDE 配合工作的 Microchip 在线仿真器。 MPLAB ICE 2000 支持 PICmicro MCU。MPLAB ICE 4000 支持 PIC18F MCU 和 dsPIC30F DSC。每个 ICE 的主要组件均为仿真器主机。一个完整的系统由仿真器主机、处理器模块、电缆和 MPLAB IDE 软件组成。

MPLAB IDE

Microchip 的集成开发环境。

MPLAB LIB30

MPLAB LIB30 归档器 / 库管理器是用于管理和组织由 MPLAB ASM30 或 MPLAB C30 C 编译器生成的 COFF 目标模块的目标库管理器。

MPLAB LINK30

MPLAB LINK30 是 Microchip MPLAB ASM30 汇编器和 Microchip MPLAB C30 C 编译器的目标链接器。

MPLAB PM3

Microchip 的器件编程器。可对 PIC18 单片机和 dsPIC 数字信号控制器进行编程。可以与 MPLAB IDE 一起使用，也可以独立使用。它将取代 PRO MATE II。

MPLAB SIM

在 MPLAB IDE 中使用的 Microchip 软件模拟器，支持 PICmicro MCU 和 dsPIC DSC 器件。

MPLIB 目标库管理器 （MPLIB Object Librarian）

MPLIB 库管理器是管理和组织由 MPASM 汇编器 （mpasm 或 mpasmwin v2.0）或 MPLAB C1X C 编译器生成的 COFF 目标模块的目标库管理器。

MPLINK 目标链接器 （MPLINK Object Linker）

MPLINK 链接器是 Microchip MPASM 汇编器和 Microchip MPLAB C17 或 C18 C 编译器的目标链接器。 MPLINK 链接器还可以与 Microchip MPLIB 库管理器一起使用。MPLINK 链接器是为在 MPLAB IDE 中使用而设计的，但是并不一定要在 MPLAB IDE 中使用它。

MRU

最近使用过的。指可从 MPLAB IDE 主下拉菜单中选择的文件和窗口。

NOP

空操作。除了让程序计数器加 1 外，没有其他作用的指令。

OTP

一次性可编程器件。使用非窗口封装的 EPROM 器件。由于 EPROM 需要紫外线才能擦除其存储器，因此只有窗口器件才可被擦除。

PC

PC 机或程序计数器。

PC 主机 （PC Host）

任何运行受支持的 Windows 操作系统的 IBM 和兼容的 PC 机。

PICmicro MCU

PICmicro 单片机 （MCU）是指所有的 Microchip 单片机系列。

PICSTART Plus

Microchip 的开发器件编程器。可对 8、 14、 28 和 40 引脚的 PICmicro 单片机进行编程。必须和 MPLAB IDE 软件一起使用。

PRO MATE II

Microchip 的器件编程器。对大多数 PICmicro 单片机以及大多数存储器和 KEELOQ 器件进行编程。可以和 MPLAB IDE 一起使用或独立使用。

PWM 信号 （PWM Signals）

脉宽调制信号。某些 PICmicro MCU 器件具有 PWM 外设。

RAM

随机访问存储器 （数据存储器）。可以任何顺序访问其中信息的存储器。

ROM

只读存储器 （程序存储器）。不能被修改的存储器。

Run （运行）

将仿真器从暂停释放的命令，允许它运行应用代码并实时更改或响应 I/O。

SFR

*参见*特殊功能寄存器 （Special Function Registers）。

Shell

MPASM 汇编器 shell 是宏汇编器的提示输入接口。有两个 MPASM 汇编器 shell：一个针对 DOS 版本，一个针对 Windows 版本。

Skew

不同时间出现在处理器总线上与指令执行相关的信息。例如，在执行前一条指令的过程中取指时，被执行的操作码出现在总线上；当实际执行该操作码时，总线上出现源数据的地址和值以及目标数据的地址。而目标数据的值在下一条指令执行时出现在总线上。跟踪缓冲器在一个操作完成时捕获总线上的这些信息。因此，一个跟踪缓冲器记录将包含三条指令的执行信息。执行一条指令时，从一条信息到另一条信息之间捕捉的周期数称为 Skew。

Skid

当使用硬件断点来暂停处理器时，在处理器暂停之前可能再执行一条或多条额外的指令。在预期断点之后执行的额外指令数称为 skid。

Step Into （跳入）

此命令与单步运行相同。 Step Into （与 Step Over 相反）随着 CALL 指令进入子程序。

Step Out （跳出）

Step Out 允许跳出当前正在执行的子程序。此命令执行子程序中剩下的代码，然后在子程序的返回地址处停止执行。

Step Over （跳过）

Step Over 允许跳过子程序。此命令执行子程序中的代码，然后在子程序的返回地址处停止执行。

当 Step Over 一条 call 指令时，下一个断点将设置在 call 指令的下一条指令处。如果由于某种原因，子程序陷入无限循环或不正确返回，下一个断点将永远执行不到。选择 Halt 来重新获得对程序执行的控制。

Watch 窗口 （Watch Window）

Watch 窗口包含在每个断点更新的观察变量的列表。

WDT

*参见*看门狗定时器 （Watchdog Timer）。

编译器 （Compiler）

将用高级语言编写的程序翻译成机器码的程序。

操作码 （Opcodes）

操作码。*参见*助记符 （Mnemonics）。

程序存储器 （Program Memory）

器件中存储指令的存储区。也可能是仿真器或软件模拟器中包含已下载的目标应用固件的存储器。

程序计数器 （Program Counter）

包含当前正在执行的指令的地址的存储单元。

触发输出 （Trigger Output）

触发输出是指可在任何地址或地址范围产生的仿真器输出信号，它独立于跟踪和断点设置。可以设置任何数量的触发输出点。

次数计数器 （Pass Counter）

每当事件 （如在某个特定地址执行指令）发生时进行递减计数的计数器。当次数计数值达到零时，对该事件做出响应。可以将次数计数器指派给中断和跟踪逻辑，也可以将它指派给 complex trigger （复杂触发）对话框中的任何连续事件。

存档 （Archive）

可重定位的目标模块的集合。通过将多个源文件汇编为目标文件，然后使用归档器将这些目标文件组合为一个库文件来创建。库文件可以与目标模块和其他库文件链接以生成可执行代码。

单步执行 （Single Step）

此命令以一次执行一条指令的方式执行代码。每条指令执行后， MPLAB IDE 会更新寄存器窗口、观察变量和状态显示，从而使您能分析和调试指令执行。也可以单步执行 C 编译器的源代码，但是 MPLAB IDE 并不是执行单条指令而是执行所有由这行高级 C 语句生成的汇编级指令。

单片机 （Microcontroller）

一种高度集成的芯片，包含 CPU、 RAM、程序存储器、 I/O 端口和定时器。

单片机模式 （Microcontroller Mode）

PIC17 和 PIC18 单片机的一种可能的程序存储器配置。在单片机模式下，只允许执行内部存储器中的指令。因此，在单片机模式下只能使用片上程序存储器。

导出 （Export）

以标准化的格式将数据从 MPLAB IDE 发出。

导入 （Import）

将数据从外部源 （例如， hex 文件）传送到 MPLAB IDE 中。

递归 （Recursion）

递归为一个已定义的函数或宏可以调用自身的概念。写递归的宏时务必小心；当没有从递归退出的出口时，很容易陷入无限循环。

地址 （Address）

标识在存储器中的位置的值。

仿真 （Emulation）

执行装入仿真存储器的软件的过程，就好象该软件是驻留在单片机上的固件一样。

仿真存储器 （Emulation Memory）

包含在仿真器内的程序存储器。

仿真器 （Emulator）

执行仿真的硬件。

仿真器系统 （Emulator System）

MPLAB ICE 2000 和 MPLAB ICE 4000 仿真器系统包括仿真器主机、处理器模块、器件适配器、电缆以及 MPLAB IDE 软件。

仿真器主机 （Pod, Emulator）

包含仿真存储器、跟踪存储器、事件和周期定时器以及跟踪 / 断点逻辑的外部仿真器机盒。

非实时 （Non Real-Time）

指处理器处于断点或正在单步执行指令或者 MPLAB IDE 正运行在软件模拟器模式下。

非易失性存储器 （Non-Volatile Storage）

关断电源后，能保存自身内容的存储设备。

符号 （Symbol）

符号是用于描述组成程序的各个部分的通用机制。这些部分包括函数名、变量名、段名、文件名、结构 / 枚举类 / 联合标记名等。 MPLAB IDE 中的符号主要指变量名、函数名和汇编标号。链接后符号的值是其在存储器中的值。

概要文件 （Profile）

对于 MPLAB SIM 软件模拟器来说，该文件是对寄存器执行的激励的摘要列表。

高级语言 （High Level Language）

用于编写比汇编语言更加脱离处理器机器码的程序的语言。

跟踪 （Trace）

记录程序执行的仿真器或软件模拟器功能。仿真器将程序执行记录到其跟踪缓冲器中，该缓存器中的内容将被上载到 MPLAB IDE 的 trace （跟踪）窗口。

跟踪存储器 （Trace Memory）

跟踪存储器包含在仿真器内。跟踪存储器有时也称为跟踪缓冲器。

工具栏 （Tool Bar）

一行或一列图标，可以在其中单击以执行 MPLAB IDE 功能。

工作簿 （Workbook）

对于 MPLAB SIM 软件模拟器来说，是一种用于产生 SCL 激励的设置。

观察变量 （Watch Variable）

调试会话期间可在 watch 窗口中监控的变量。

归档器 （Archiver）

创建和管理库的工具。

国际标准化组织 （International Organization for Standardization）

在很多业务和技术方面 （包括计算和通信）设置标准的组织。

宏 （Macro）

宏指令。以缩写形式表示一系列指令的指令。

宏伪指令 （Macro Directives）

在宏体定义内部控制执行和数据分配的伪指令。

环境——IDE （Environment – IDE）

用于开发应用程序的特定桌面布局。

环境——MPLAB PM3 （Environment – MPLAB PM3）

包含有关如何对器件编程的文件的文件夹。此文件夹可被传输到一个 SD/MMC 卡。

汇编器 （Assembler）

将汇编语言源代码翻译成机器码的语言工具。

汇编语言 （Assembly Language）

用符号形式描述二进制机器码的编程语言。

激励 （Stimulus）

软件模拟器的输入，即，为模拟对外部信号的响应而生成的数据。通常这些数据以动作列表的形式放在文本文件中。激励可能是异步、同步 （引脚）、时钟激励和寄存器激励。

机器码 （Machine Code）

处理器实际读取和解释的计算机程序表示。用二进制机器码编写的程序由一系列机器指令组成 （可能还带有数据）。某个特定处理器的所有可用指令的集合称为它的 “指令集”。

机器语言 （Machine Language）

用于特定中央处理单元的一组指令，旨在无需翻译即可由处理器使用。

基数 （Radix）

数基，十六进制或十进制，用于指定地址。

交叉引用文件 （Cross Reference File）

引用符号表的文件和引用符号的文件列表。如果符号已被定义，那么列出的第一个文件是该定义的位置。剩余的文件包含对该符号的引用。

校准存储区 （Calibration Memory）

用来保存 PICmicro 单片机的片上 RC 振荡器或其他器件外设的校准值的特殊功能寄存器。

节点 （Node）

MPLAB IDE 项目组件。

警告 （Warning）

向您提供的警告，以在将对器件、软件文件或设备造成物理损害的情形下发出警告。

静态 RAM 或 SRAM （Static RAM or SRAM）

静态随机访问存储器。可在目标板上读写的程序存储器，无需经常刷新。

局部标号 （Local Label）

是用 LOCAL 伪指令在宏内部定义的标号。这些标号特定于宏实例化的一个给定实例。换句话说，当遇到 ENDM 宏之后，将不再能访问声明为 local 的符号和标号。

绝对段 （Absolute Section）

具有固定 （绝对）地址且不能被链接器更改的段。

看门狗定时器 （Watchdog Timer）

PICmicro 单片机上的一个定时器，它可在可选的时间长度之后复位处理器。可通过使用配置位使能、禁止或设置 WDT。

控制伪指令 （Control Directives）

汇编语言代码中的伪指令，它能根据指定表达式的汇编时值包含或省略代码。

库 （Library）

参见存档 （Archive）。

库管理器 （Librarian）

*参见*归档器 （Archiver）。

快速操作存储器 （仅 PIC18）（Access Memory (PIC18 Only)）

无论存储区选择寄存器 （bank select register， BSR）的设置如何都允许访问的 PIC18 特殊寄存器。

扩展的单片机模式 （Extended Microcontroller Mode）

在扩展的单片机模式下，片上程序存储器和外部存储器都可用。如果程序存储器的地址超出了 PIC17 或 PIC18 器件的内部存储器空间，程序执行将自动切换到外部存储器。

链接器 （Linker）

通过解析从一个模块到另一个模块的引用，将目标文件和库相结合以生成可执行代码的语言工具。

链接描述文件 （Linker Script Files）

链接描述文件是链接器的命令文件。它们定义链接器选项并描述目标平台上的可用存储器。

列表伪指令 （Listing Directives）

列表伪指令是那些控制汇编器列表文件格式的伪指令。它们实现了标题、分页和其他列表控制的规范。

列表文件 （Listing File）

列表文件是一种 ASCII 文本文件，它显示了为每个 C 源语句、汇编指令、汇编器伪指令或在源文件中遇到的宏生成的机器码。

逻辑探头 （Logic Probes）

至多 14 个逻辑探头可被连接到某些 Microchip 仿真器。逻辑探头提供外部跟踪输入、触发输出信号、 +5V 电压以及一个公共接地端。

命令行接口 （Command Line Interface）

程序及其用户之间的一种完全基于文本输入和输出的通信手段。

模板 （Template）

建立的文本行，用于在以后插入到文件中。MPLAB 编辑器将模板存储在模板文件中。

目标 （Target）

指用户硬件。

目标板 （Target Board）

电路和组成目标应用的可编程器件。

目标处理器 （Target Processor）

目标应用板上的单片机器件。

目标代码 （Object Code）

由汇编器或编译器生成的机器码。

目标文件 （Object File）

包含机器码 （可能还有调试信息）的文件。它可被立即执行或者也可能需要与其他目标文件 （例如，库文件）链接重定位，以生成完整的可执行程序。

目标文件伪指令 （Object File Directives）

仅当创建目标文件时使用的伪指令。

目标应用程序 （Target Application）

驻留在目标板上的软件。

内部链接 （Internal Linkage）

如果某个函数或变量不可以从定义它的模块外被访问，则称该函数或变量有内部链接。

跑表 （Stopwatch）

测量执行周期的计数器。

配置位 （Configuration Bits）

可通过编程来设置 PICmicro 单片机工作模式的专用位。配置位可能预编程也可能不预编程。

片外存储器 （Off-Chip Memory）

片外存储器的使用由 PIC17 或 PIC18 器件的存储器选择选项指定，存储器可驻留在目标板上，也可由仿真器提供所有的程序存储器。

器件编程器 （Device Programmer）

用来对可电编程的半导体器件 （如单片机）进行编程的工具。

嵌套深度 （Nesting Depth）

一个宏可以包含其他宏的最大级数。

情形 （Scenario）

对于 MPLAB SIM 软件模拟器来说，是一种用于激励控制的特定设置。

软件断点 （Breakpoint, Software）

固件执行将停止的地址。通常由特殊的中断指令实现。

软件堆栈 （Stack, Software）

供应用程序用来存储返回地址、函数参数和局部变量的存储区。当用高级语言开发代码时，通常由编译器管理此存储区。

软件模拟器 （Simulator）

模拟器件操作的软件程序。

筛选 （Filter）

通过选择确定在跟踪显示或数据文件中将包含 / 排除哪些数据。

闪存 （Flash）

一种 EEPROM，其数据以块形式而不是以字节形式写入或擦除。

上电复位仿真 （Power-on-Reset Emulation）

在数据 RAM 区域中写入随机值以模拟最初给应用上电时 RAM 中未初始化的值的软件随机化过程。

上载 （Upload）

上载功能将数据从一个工具 （如仿真器或编程器）传输到主机 PC 或者从目标板传输到仿真器。

实时 （Real-Time）

在仿真器或 MPLAB ICD 模式下，当从暂停状态释放时，处理器将在实时模式下运行并且完全像正常芯片那样工作。在实时模式下， MPLAB ICE 的实时跟踪缓冲器被使能并不断地捕获所有选定的周期，同时所有的中断逻辑也被使能。在仿真器或 MPLAB ICD 中，处理器将实时执行直到由于遇到有效的断点而暂停，或者直到用户暂停仿真器。对于软件模拟器，实时只不过意味着单片机指令的执行速度与主机 CPU 模拟这些指令的速度相同。

事件 （Event）

对总线周期的描述，可能包括地址、数据、次数计数、外部输入、周期类型 （取指或读 / 写）以及时间标记。事件可用来描述触发、断点和中断。

数据存储器 （Data Memory）

在 Microchip MCU 和 DSC 器件上，数据存储器 （RAM）由通用寄存器 （general purpose register， GPR）和特殊功能寄存器 （special function register， SFR）组成。有些器件还具有 EEPROM 数据存储器。

数据伪指令 （Data Directives）

数据伪指令是那些控制汇编器对程序和数据存储器进行分配的指示性语句，它提供了用符号 （即有意义的名称）引用数据项的方法。

数字信号控制器 （Digital Signal Controller）

具有数字信号处理功能的单片机器件，即 Microchip dsPIC 器件。

特殊功能寄存器 （Special Function Registers）

为数据存储器 （RAM）的一部分，由专用于控制 I/O 处理功能、 I/O 状态、定时器或者其他模式或外设的寄存器组成。

外部 RAM （External RAM）

片外读 / 写存储器。

外部标号 （External Label）

有外部链接的标号。

外部符号 （External Symbol）

有外部链接的标识符符号。它可能是一个引用，也可能是一个定义。

外部符号解析 （External Symbol Resolution）

由链接器执行的过程，在该过程中将收集来自所有输入模块的外部符号定义以尝试解析所有的外部符号引用。任何没有对应定义的外部符号引用将会导致报告发生链接器错误。

外部链接 （External Linkage）

如果某个函数或变量可以从定义它的模块外被引用，则称该函数或变量有外部链接。

外部输入线 （External Input Line）

用于根据外部信号设置事件的外部输入信号逻辑探测线 （TRIGIN）。

微处理器模式 （Microcontroller Mode）

PIC17 和 PIC18 单片机的一种可能的程序存储器配置。在微处理器模式下，不使用片上程序存储器。整个程序存储器映射到外部。

伪指令 （Directives）

源代码中提供对语言工具的操作进行控制的语句。

未初始化的数据 （Uninitialized Data）

未定义初始值的数据。如在 C 中的：

```
int myVar;
```

定义了一个将驻留在未初始化数据段中的变量。

文件寄存器 （File Register）

片上数据存储器包括通用寄存器 （GPR）和特殊功能寄存器 （SFR）。

系统窗口控件 （System Window Control）

系统窗口控件位于窗口和某些对话框的左上角。单击该控件通常会弹出一个含有 Minimize （最小化）、 Maximize （最大化）和 Close （关闭）项的菜单。

下载 （Download）

下载是将数据从主机传送到另一个器件 （如仿真器、编程器或目标板）的过程。

限定符 （Qualifier）

次数计数器使用的地址或地址范围，或者为在复杂触发中发生在另一个操作前的事件。

项目 （Project）

为应用程序编译目标和可执行代码的一组源文件和指令。

消息 （Message）

警告您语言工具操作中存在潜在问题的文本。消息并不会停止语言工具的操作。

异步激励 （Asynchronous Stimulus）

为模拟到被模拟器件的外部输入而生成的数据。

应用 （Application）

可以由 PICmicro 单片机控制的一组软件和硬件。

硬件断点 （Breakpoint, Software）

执行后将导致程序暂停的事件。

硬件堆栈 （Stack, Hardware）

当调用函数时，在 PICmicro 单片机中存储返回地址的存储单元。

源代码 （Source Code）

由编程人员编写的计算机程序的形式。源代码以某种正规的编程语言编写，可被翻译成机器码或由解释程序执行。

原始数据 （Raw Data）

与段相关的代码或数据的二进制表示。

原型系统 （Prototype System）

指用户的目标应用或目标板的术语。

源文件 （Source File）

包含源代码的 ASCII 文本文件。

运算符 （Operators）

加号 “+”和减号 “-”之类的符号，它们在构成定义明确的表达式时使用。每个运算符都有已分配的优先级，用来决定求值的顺序。

暂停 （Halt）

停止程序执行。执行暂停与在断点停止相同。

指令 （Instructions）

告诉中央处理单元执行某个特定操作的位序列，这些位可以包含在该操作中将使用的数据。

指令集 （Instruction Set）

某个特定处理器能够理解的机器语言指令的集合。

中断 （Interrupt）

中断为一个到 CPU 的信号，该信号可暂停执行正在运行的应用程序并将控制转移给中断服务程序 （Interrupt Service Routine， ISR）从而能对该事件进行处理。

中断处理程序 （Interrupt Handler）

发生中断时，处理特殊代码的程序。

中断服务程序 （Interrupt Service Routine）

用户产生的代码，当发生中断时进入该程序。程序存储器中代码的位置通常取决于已发生的中断的类型。

中断请求 （Interrupt Request）

使处理器暂时挂起正常的指令执行并开始执行中断处理程序的事件。有些处理器具备几个中断请求事件，以允许产生不同优先级的中断。

助记符 （Mnemonics）

可以直接翻译成机器码的文本指令。也称为操作码。

状态栏 （Status Bar）

状态栏位于 MPLAB IDE 窗口的底部，指示诸如光标位置、开发模式、器件和活动工具栏之类的当前信息。

字母数字 （Alphanumeric）

字母数字字符由字母字符和十进制数字 （0、 1、……、 9）组成。

字母字符 （Alphabetic Character）

字母字符是指那些在阿拉伯字母表中的字母字符
（a、 b、……、 z， A、 B、……、 Z）。

注：



索引

注：



全球销售及服务网点

美洲

公司总部 Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277
技术支持：
http://support.microchip.com
网址：www.microchip.com

亚特兰大 Atlanta
Alpharetta, GA
Tel: 1-770-640-0034
Fax: 1-770-640-0307

波士顿 Boston
Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 Chicago
Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

达拉斯 Dallas
Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 Detroit
Farmington Hills, MI
Tel: 1-248-538-2250
Fax: 1-248-538-2260

科科莫 Kokomo
Kokomo, IN
Tel: 1-765-864-8360
Fax: 1-765-864-8387

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608

圣何塞 San Jose
Mountain View, CA
Tel: 1-650-215-1444
Fax: 1-650-961-0286

加拿大多伦多 Toronto
Mississauga, Ontario,
Canada
Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

中国 – 北京
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

中国 – 成都
Tel: 86-28-8676-6200
Fax: 86-28-8676-6599

中国 – 福州
Tel: 86-591-8750-3506
Fax: 86-591-8750-3521

中国 – 香港特别行政区
Tel: 852-2401-1200
Fax: 852-2401-3431

中国 – 青岛
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 – 上海
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 – 沈阳
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 – 深圳
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

中国 – 顺德
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

中国 – 武汉
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 – 西安
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

台湾地区 – 高雄
Tel: 886-7-536-4818
Fax: 886-7-536-4803

台湾地区 – 台北
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

台湾地区 – 新竹
Tel: 886-3-572-9526
Fax: 886-3-572-6459

亚太地区

澳大利亚 Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

印度 India - Bangalore
Tel: 91-80-2229-0061
Fax: 91-80-2229-0062

印度 India - New Delhi
Tel: 91-11-5160-8631
Fax: 91-11-5160-8632

印度 India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

日本 Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

韩国 Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 或
82-2-558-5934

马来西亚 Malaysia - Penang
Tel:604-646-8870
Fax:604-646-5086

菲律宾 Philippines - Manila
Tel: 011-632-634-9065
Fax: 011-632-634-9069

新加坡 Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

泰国 Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

欧洲

奥地利 Austria - Weis
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

丹麦 Denmark-Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

法国 France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

意大利 Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

荷兰 Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

西班牙 Spain - Madrid
Tel: 34-91-352-30-52
Fax: 34-91-352-11-47

英国 UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

07/01/05